

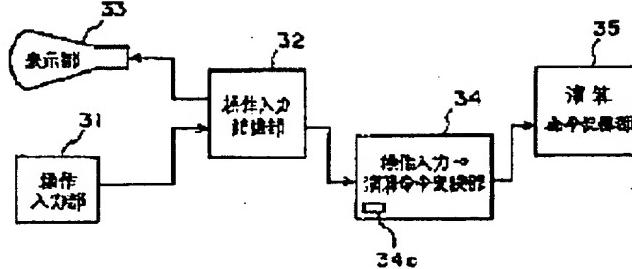
## PROGRAMMING DEVICE

**Patent number:** JP7168707  
**Publication date:** 1995-07-04  
**Inventor:** MATSUDA SHIGEHIKO  
**Applicant:** MITSUBISHI ELECTRIC CORP  
**Classification:**  
 - international: G06F9/06  
 - european:  
**Application number:** JP19930316595 19931216  
**Priority number(s):** JP19930316595 19931216

[Report a data error here](#)

### Abstract of JP7168707

**PURPOSE:** To realize batch high-speed processing for the data of the same arithmetic processing so as to reduce overhead time when executing arithmetic for plural data by resolving a data flow chart, which is prepared on the screen of a CRT, for each of common arithmetic processing for the data and preparing a program group to execute that arithmetic. **CONSTITUTION:** A program preparing means 34a resolves the data flow chart, which is prepared on the screen of a CRT 33 by a pointing device 31, for each of common arithmetic processing to the data and prepares the program group to execute that arithmetic. The program preparing means 34a decides the kind of arithmetic such as 'average processing' by scanning the screen from left to right or from up to down. Next, an input/output variable for each of arithmetic processing and the number of those variables are provided. Then, the declaration sentence part of the input/output variable is prepared. Further, the repeating part of processing is prepared. Afterwards, the declaration sentence part of the input/output variable is prepared and moreover, the repeating part of processing is prepared. Then, an arithmetic processing part is prepared. Thus, the program of arithmetic concerning 'average processing' is prepared.



Data supplied from the **esp@cenet** database - Worldwide

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-168707

(43) 公開日 平成7年(1995)7月4日

(51) Int.Cl.<sup>6</sup>

G 06 F 9/06

識別記号 庁内整理番号

530 H 9367-5B

F I

技術表示箇所

(21) 出願番号 特願平5-316595

(22) 出願日 平成5年(1993)12月16日

審査請求 未請求 請求項の数35 O.L (全28頁)

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 松田 茂彦

神戸市兵庫区和田崎町1丁目1番2号 三菱電機株式会社制御製作所内

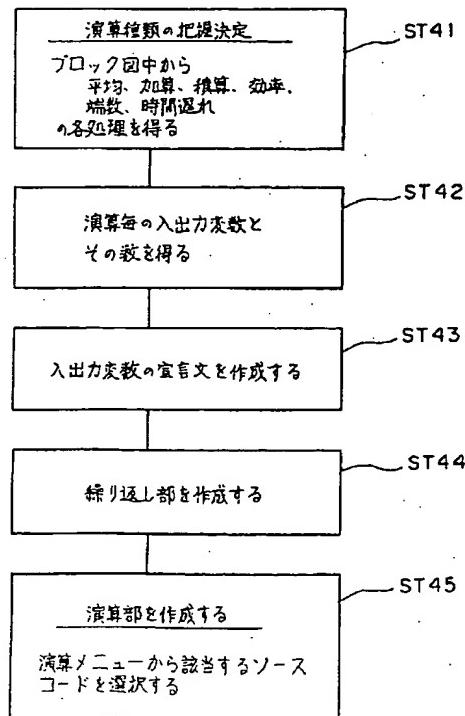
(74) 代理人 弁理士 田澤 博昭 (外1名)

(54) 【発明の名称】 プログラミング装置

(57) 【要約】

【目的】 同一演算処理のデータに対し同一演算を一括して実行する一括高速処理を実現し、複数のデータに対する演算実行時のオーバーヘッド時間を削減する。

【構成】 データに対する演算処理をブロック図化したデータフロー図として作成するためのCRT33およびポインティングデバイスと、上記CRT33の画面上に作成されたデータフロー図をデータに対する共通の演算処理毎に分解し、その演算を実施するプログラム群を作成するプログラム作成手段34aとを備える。



## 【特許請求の範囲】

【請求項1】 データに対する演算処理をブロック図化したデータフロー図として作成するためのCRTおよびポインティングデバイスと、上記CRTの画面上に作成されたデータフロー図をデータに対する共通の演算処理毎に分解し、その演算を実施するプログラム群を作成するプログラム作成手段とを備えたプログラミング装置。

【請求項2】 データに対する演算処理をブロック図化したデータフロー図として作成するためのCRTと、ポインティングデバイスと、上記CRTの画面上に作成されたデータフロー図をデータに対する共通の演算処理毎に分解し、その演算を実施するプログラム群を作成するプログラム作成手段と、そのプログラム作成手段において上記プログラム群を作成するのと同時に、そのプログラムのための入出力データファイルを生成する入出力データファイル自動生成手段とを備えたプログラミング装置。

【請求項3】 データに対する演算処理をブロック図化したデータフロー図として作成するためのCRTと、ポインティングデバイスと、上記CRTの画面上に作成されたデータフロー図をデータに対する共通の演算処理毎に分解し、その演算を実施するプログラム群を作成するプログラム作成手段と、そのプログラム作成手段において上記プログラム群を作成するのと同時に、そのプログラムのための実行タイミングをスケジューリングするスケジューリング手段とを備えたプログラミング装置。

【請求項4】 データに対する演算処理をブロック図化したデータフロー図として作成するためのCRTと、ポインティングデバイスと、上記CRTの画面上に作成されたデータフロー図をデータに対する共通の演算処理毎に分解し、その演算を実施するプログラム群を作成するプログラム作成手段と、そのプログラム作成手段において上記プログラム群を作成するのと同時に、そのプログラムのための入出力データファイルを生成する入出力データファイル自動生成手段と、そのプログラムのための実行タイミングをスケジューリングするスケジューリング手段とを備えたプログラミング装置。

【請求項5】 元のデータフロー図上で発生した演算処理内容変更に対する処理を、出力されるプログラム上の変更箇所に対応する部分のみ修正し出力する変更箇所修正出力手段を備えた請求項1記載のプログラミング装置。

【請求項6】 元のデータフロー図上で発生した演算処理内容変更に対する処理を、入出力データファイル上の変更箇所のみ修正し出力する変更箇所修正出力手段を備えた請求項2記載のプログラミング装置。

【請求項7】 元のデータフロー図上で発生した演算処理内容変更に対する処理を、スケジューリング手段によりスケジューリングしたスケジュール上の変更箇所のみ修正し出力する変更箇所修正出力手段を備えた請求項

## 3記載のプログラミング装置。

【請求項8】 データに対する演算処理をブロック図化したデータフロー図を表わした手書きOCRシートと、その手書きOCRシートに表わされたデータフロー図をデータに対する共通の演算処理毎に分解し、その演算を実施するプログラム群を作成するプログラム作成手段とを備えた請求項1から請求項7のうちいずれか1項記載のプログラミング装置。

【請求項9】 複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段を備えたプログラミング装置。

【請求項10】 複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリントに出力するデータフロー可視化手段を備えたプログラミング装置。

【請求項11】 プログラミングにおけるデータの演算順序の連続性を判定する連続性判定手段を備えた請求項9または請求項10記載のプログラミング装置。

【請求項12】 出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示する演算順序／演算タイミング同時表示手段を備えた請求項9または請求項10記載のプログラミング装置。

【請求項13】 元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上に変更箇所のみ追加出力する変更箇所追加出力手段を備えた請求項9または請求項10記載のプログラミング装置。

【請求項14】 元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図上での変更箇所がない場合に修正出力処理を行なわない修正出力制御手段を備えた請求項9または請求項10記載のプログラミング装置。

【請求項15】 複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段を備えた請求項1から請求項7のうちいずれか1項記載のプログラミング装置。

【請求項16】 複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段を備えた請求項8記載のプログラミング装置。

【請求項17】 複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリントに出力するデータフロー可視化手段を備えた請求項1から請求

項7のうちいずれか1項記載のプログラミング装置。

【請求項18】複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリンタに出力するデータフロー可視化手段を備えた請求項8記載のプログラミング装置。

【請求項19】複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段と、プログラミングにおけるデータの演算順序の連続性を判定する連続性判定手段とを備えた請求項1から請求項7のうちいずれか1項記載のプログラミング装置。

【請求項20】複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段と、プログラミングにおけるデータの演算順序の連続性を判定する連続性判定手段とを備えた請求項8記載のプログラミング装置。

【請求項21】複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段と、出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示する演算順序／演算タイミング同時表示手段とを備えた請求項1から請求項7のうちいずれか1項記載のプログラミング装置。

【請求項22】複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段と、出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示する演算順序／演算タイミング同時表示手段とを備えた請求項8記載のプログラミング装置。

【請求項23】複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段と、元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上に変更箇所のみ追加出力する変更箇所追加出力手段とを備えた請求項1から請求項7のうちいずれか1項記載のプログラミング装置。

【請求項24】複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段と、元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上に変更箇所のみ追加出力する変更箇所追加出力手段とを備えた請求項8記載のプログラミング装置。

【請求項25】複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段と、元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図上での変更箇所がない場合に修正出力処理を行なわない修正出力制御手段とを備えた請求項1から請求項7のうちいずれか1項記載のプログラミング装置。

【請求項26】複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段と、元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図上での変更箇所がない場合に修正出力処理を行なわない修正出力制御手段とを備えた請求項8記載のプログラミング装置。

【請求項27】複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリンタに出力するデータフロー可視化手段と、プログラミングにおけるデータの演算順序の連続性を判定する連続性判定手段とを備えた請求項1から請求項7のうちいずれか1項記載のプログラミング装置。

【請求項28】複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリンタに出力するデータフロー可視化手段と、プログラミングにおけるデータの演算順序の連続性を判定する連続性判定手段とを備えた請求項8記載のプログラミング装置。

【請求項29】複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリンタに出力するデータフロー可視化手段と、出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示する演算順序／演算タイミング同時表示手段とを備えた請求項1から請求項7のうちいずれか1項記載のプログラミング装置。

【請求項30】複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリンタに出力するデータフロー可視化手段と、出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示する演算順序／演算タイミング同時表示手段とを備えた請求項8記載のプログラミング装置。

【請求項31】複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリンタに出力するデータフロー可視化手段と、元のプログラム内で発

生した演算対象データの変更に対する処理を出力データフロー図上に変更箇所のみ追加出力する変更箇所追加出力手段とを備えた請求項1から請求項7のうちいずれか1項記載のプログラミング装置。

【請求項32】複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリンタに出力するデータフロー可視化手段と、元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上に変更箇所のみ追加出力する変更箇所追加出力手段とを備えた請求項8記載のプログラミング装置。

【請求項33】複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリンタに出力するデータフロー可視化手段と、元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図上での変更箇所がない場合に修正出力処理を行なわない修正出力制御手段とを備えた請求項1から請求項7のうちいずれか1項記載のプログラミング装置。

【請求項34】複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリンタに出力するデータフロー可視化手段と、元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図上での変更箇所がない場合に修正出力処理を行なわない修正出力制御手段とを備えた請求項8記載のプログラミング装置。

【請求項35】プログラムを実行する装置と一体的に構成した請求項1から請求項15または請求項17記載のプログラミング装置。

#### 【発明の詳細な説明】

##### 【0001】

【産業上の利用分野】この発明は、プログラムの作成や修正を容易に行なうことのできる、計算機のプログラミング装置に関するものである。

##### 【0002】

【従来の技術】図26は例えば、特公昭64-8361号公報に示された従来のプログラミング装置の表示部に表示される論理回路図である。図27は、図26に示す論理回路図を表示する表示部や図28に示すキー入力部を有したプログラミング装置の構成図であり、図において31は図26に示す論理回路図などを入力するキー入力部、32はキー入力記憶部、33は図26に示した論理回路図などを表示するCRTあるいは液晶表示装置あるいはプラズマディスプレイ装置などの表示部、34はキー入力→シーケンス命令変換部、35はシーケンス命令記憶部、36はシーケンス命令→キー入力変換部である。図28は、図27に示したプログラミング装置の表

示部33に表示される論理回路図を作成するために用いるキー入力部のキー配列を示している。図29は、論理積(AND)や論理和(OR)、反転(NOT)の論理回路図などとその論理回路図などに対応した命令変換規則を示す論理記号説明図である。

【0003】次に動作について説明する。キー入力部31により図26に示す論理回路図が入力されると、この論理回路図は入力部37と分岐部38と論理演算部39と出力部40に分れる。ここで論理演算部39は、図29に示す論理積(AND)や論理和(OR)、反転(NOT)である。これら論理積や論理和や反転を示す論理回路図は図27のキー入力記憶部32に格納され、図29の論理記号説明図に示す命令変換規則によりシーケンス命令に変換され、シーケンス命令記憶部35に格納される。このときの演算の処理順序は、図30に示すように自動的に決定される。

##### 【0004】

【発明が解決しようとする課題】従来のプログラミング装置は以上のように構成されているので、演算処理は図26の論理回路図に代表されるようなデータフロー図から一意的に決定され、通常計算機が得意とする同一演算対象データの一括高速処理を適用することが困難であり、演算内容の変更に伴うオーバーヘッド時間、演算毎に異なる入出力データの種類や数の変更に伴うオーバーヘッド時間を無駄に費やしてしまう問題点があった。

【0005】請求項1の発明は上記のような問題点を解消するためになされたもので、同一演算処理のデータに対し同一演算を一括して実行する一括高速処理を実現し、複数のデータに対する演算実行時のオーバーヘッド時間を削減できるプログラミング装置を得ることを目的とする。

【0006】請求項2の発明は、同一演算処理のデータに対し同一演算を一括して実行するプログラムの生成と同時に、そのプログラムのための入出力データのファイルを自動生成し、前記プログラムと入出力データのファイルとを分離できるプログラミング装置を得ることを目的とする。

【0007】請求項3の発明は、同一演算処理のデータに対し同一演算を一括して実行するプログラムの実行タイミングを自動的にスケジューリングし、人為的な処理順序の誤りを防止できるプログラミング装置を得ることを目的とする。

【0008】請求項4の発明は、同一演算処理のデータに対し同一演算を一括して実行するプログラムの生成と同時に、そのプログラムのための入出力データファイルを自動生成すると共に、上記プログラムのための実行タイミングを自動的にスケジューリングできる、一括高速処理を実現すると共に信頼性を向上させたプログラミング装置を得ることを目的とする。

##### 【0009】請求項5の発明は、元のデータフロー図上

で発生した演算処理内容変更に伴って生ずる時間や資料の無駄を省くことのできるプログラミング装置を得ることを目的とする。

【0010】請求項6の発明は、元のデータフロー図上で発生した演算処理内容変更に対し、入出力データファイル上の変更箇所のみ修正し出力することで元のデータフロー図上で発生した演算処理内容変更に伴って生ずる時間や資料の無駄を省くことのできるプログラミング装置を得ることを目的とする。

【0011】請求項7の発明は、元のデータフロー図上で発生した演算処理内容変更に対し、スケジュール上で変更箇所のみ修正し出力できるプログラミング装置を得ることを目的とする。

【0012】請求項8の発明は、データフロー図を表わした手書きOCRシートを用いることで、データフロー図の作成を容易にするプログラミング装置を得ることを目的とする。

【0013】請求項9の発明は、変更後のプログラムをデータフロー図としてCRT画面に可視化することで変更結果を直接目視によりチェックできるプログラミング装置を得ることを目的とする。

【0014】請求項10の発明は、変更後のプログラムをデータフロー図としてプリンタに出力することで、上記データフロー図を出力する際の表示範囲の制限に関する障害を排除できるプログラミング装置を得ることを目的とする。

【0015】請求項11の発明は、プログラムを変更したときの誤りを検出できるプログラミング装置を得ることを目的とする。

【0016】請求項12の発明は、演算順序の逆転などの発生の有無を目視により容易に判定できるようになることでプログラムの修正の際の誤りのチェックを容易にするプログラミング装置を得ることを目的とする。

【0017】請求項13の発明は、元のプログラム内で発生した演算対象データを変更したときに、それに従つて行なわれる修正処理時間や修正される資料の無駄を省くことのできるプログラミング装置を得ることを目的とする。

【0018】請求項14の発明は、元のプログラム内における変更が、他に与える影響が小さい場合にその変更による修正処理に要する時間を省き、修正処理に要する時間を削減できるプログラミング装置を得ることを目的とする。

【0019】請求項15の発明は、データフロー図とプログラム間の変換を双方向にすると共に、上記データフロー図をCRTに出力できるプログラミング装置を得ることを目的とする。

【0020】請求項16の発明は、データフロー図とプログラム間の変換を双方向にすると共に、データフロー図を入力するのにそのデータフロー図を表わした手書き

OCRシートを用いることができ、さらに上記データフロー図をCRTに出力できるプログラミング装置を得ることを目的とする。

【0021】請求項17の発明は、データフロー図とプログラム間の変換を双方向にすると共に、上記データフロー図をプリンタに出力できるプログラミング装置を得ることを目的とする。

【0022】請求項18の発明は、データフロー図とプログラム間の変換を双方向にすると共に、データフロー図を入力するのに手書きOCRシートを用いることができ、さらに上記データフロー図をプリンタに出力できるプログラミング装置を得ることを目的とする。

【0023】請求項19の発明は、CRTを用いることができプログラムの作成／変更を容易にすると共にプログラムの作成の際の誤りを容易に検出できるプログラミング装置を得ることを目的とする。

【0024】請求項20の発明は、CRTと手書きOCRシートを用いることができ、プログラムの作成／変更を容易にすると共にプログラムの作成の際の誤りを容易に検出できるプログラミング装置を得ることを目的とする。

【0025】請求項21の発明は、CRTを用いることができプログラムの作成／変更を容易にすると共に演算順序の逆転などの発生の有無を目視により容易に判定できるプログラミング装置を得ることを目的とする。

【0026】請求項22の発明は、CRTと手書きOCRシートを用いることができプログラムの作成／変更を容易にすると共に演算順序の逆転などの発生の有無を目視により容易に判定できるプログラミング装置を得ることを目的とする。

【0027】請求項23の発明は、CRTを用いることができプログラムの作成／変更を容易にすると共に、元のプログラム内で発生した演算対象データを変更したときに、それに従つて行なわれる修正処理時間や修正される資料の無駄を省くことのできるプログラミング装置を得ることを目的とする。

【0028】請求項24の発明は、CRTと手書きOCRシートを用いることができプログラムの作成／変更を容易にすると共に、元のプログラム内で発生した演算対象データを変更したときに、それに従つて行なわれる修正処理時間や修正される資料の無駄を省くことのできるプログラミング装置を得ることを目的とする。

【0029】請求項25の発明は、CRTを用いることができプログラムの作成／変更を容易にすると共に、元のプログラム内における変更が、出力されるデータフロー図の修正にならないときにはその修正処理を行なわないことで、上記プログラム内における変更に対する処理時間を削減できるプログラミング装置を得ることを目的とする。

【0030】請求項26の発明は、CRTと手書きOCR

Rシートを用いることができプログラムの作成／変更を容易にすると共に、元のプログラム内における変更が、出力されるデータフロー図の修正にならないときにはその修正処理を行なわないことで、上記プログラム内における変更に対する処理時間を削減できるプログラミング装置を得ることを目的とする。

【0031】請求項27の発明は、プリンタを用いることができプログラムの作成／変更を容易にすると共にプログラムの作成の際の誤りを容易に検出できるプログラミング装置を得ることを目的とする。

【0032】請求項28の発明は、手書きOCRシートとプリンタを用いることができプログラムの作成／変更を容易にすると共にプログラムの作成の際の誤りを容易に検出できるプログラミング装置を得ることを目的とする。

【0033】請求項29の発明は、プリンタを用いることができプログラムの作成／変更を容易にすると共に演算順序の逆転などの発生の有無を目視により容易に判定できるプログラミング装置を得ることを目的とする。

【0034】請求項30の発明は、手書きOCRシートとプリンタを用いることができプログラムの作成／変更を容易にすると共に演算順序の逆転などの発生の有無を目視により容易に判定できるプログラミング装置を得ることを目的とする。

【0035】請求項31の発明は、プリンタを用いることができプログラムの作成／変更を容易にすると共に、元のプログラム内で発生した演算対象データを変更したときに、それに従って行なわれる修正処理時間や修正される資料の無駄を省くことのできるプログラミング装置を得ることを目的とする。

【0036】請求項32の発明は、手書きOCRシートとプリンタを用いることができプログラムの作成／変更を容易にすると共に、元のプログラム内で発生した演算対象データを変更したときに、それに従って行なわれる修正処理時間や修正される資料の無駄を省くことのできるプログラミング装置を得ることを目的とする。

【0037】請求項33の発明は、プリンタを用いることができプログラムの作成／変更を容易にすると共に、元のプログラム内における変更が、出力されるデータフロー図の修正にならないときにはその修正処理を行なわないことで、上記プログラム内における変更に対する処理時間を削減できるプログラミング装置を得ることを目的とする。

【0038】請求項34の発明は、手書きOCRシートとプリンタを用いることができプログラムの作成／変更を容易にすると共に、元のプログラム内における変更が、出力されるデータフロー図の修正にならないときにはその修正処理を行なわないことで、上記プログラム内における変更に対する処理時間を削減できるプログラミング装置を得ることを目的とする。

【0039】請求項35の発明は、プログラムの作成／修正後のプログラムロードに要する時間を削減できるプログラミング装置を得ることを目的とする。

【0040】

【課題を解決するための手段】請求項1の発明に係るプログラミング装置は、CRTの画面上にポイントティングデバイスにより作成されたデータフロー図をデータに対する共通の演算処理毎に分解し、その演算を実施するプログラム群を作成するプログラム作成手段を備えたものである。

【0041】請求項2の発明に係るプログラミング装置は、共通の演算処理を実施するプログラム群を作成するのと同時に、そのプログラムのための入出力データファイルを生成する入出力データファイル自動生成手段を備えたものである。

【0042】請求項3の発明に係るプログラミング装置は、共通の演算処理を実施するプログラム群を作成するのと同時に、そのプログラムの実行タイミングをスケジューリングするスケジューリング手段を備えたものである。

【0043】請求項4の発明に係るプログラミング装置は、共通の演算処理を実施するプログラム群を作成するのと同時に、そのプログラムのための入出力データファイルを生成する入出力データファイル自動生成手段および、上記プログラムの実行タイミングをスケジューリングするスケジューリング手段を備えたものである。

【0044】請求項5の発明に係るプログラミング装置は、元のデータフロー図上で発生した演算処理内容変更に対する処理を、出力されるプログラム上の変更箇所に対応する部分のみ修正し出力する変更箇所修正出力手段を備えたものである。

【0045】請求項6の発明に係るプログラミング装置は、元のデータフロー図上で発生した演算処理内容変更に対する処理を、入出力データファイル上の変更箇所のみ修正し出力する変更箇所修正出力手段を備えたものである。

【0046】請求項7の発明に係るプログラミング装置は、元のデータフロー図上で発生した演算処理内容変更に対する処理を、スケジューリング手段によりスケジューリングしたスケジュール上での変更箇所のみ修正し出力する変更箇所修正出力手段を備えたものである。

【0047】請求項8の発明に係るプログラミング装置は、データに対する演算処理をブロック化したデータフロー図を表わした手書きOCRシートと、その手書きOCRシートに表わされたデータフロー図をデータに対する共通の演算処理毎に分解し、その演算を実施するプログラム群を作成するプログラム作成手段とを備えたものである。

【0048】請求項9の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するブ

ログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段を備えたものである。

【0049】請求項10の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリンタに出力するデータフロー可視化手段を備えたものである。

【0050】請求項11の発明に係るプログラミング装置は、プログラミングにおけるデータの演算順序の連續性を判定する連續性判定手段を備えたものである。

【0051】請求項12の発明に係るプログラミング装置は、出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示する演算順序／演算タイミング同時表示手段を備えたものである。

【0052】請求項13の発明に係るプログラミング装置は、元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上に変更箇所のみ追加出力する変更箇所追加出力手段を備えたものである。

【0053】請求項14の発明に係るプログラミング装置は、元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図上での変更箇所がない場合に修正出力処を行なわない修正出力制御手段を備えたものである。

【0054】請求項15の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段を備えたものである。

【0055】請求項16の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段を備えたものである。

【0056】請求項17の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリンタに出力するデータフロー可視化手段を備えたものである。

【0057】請求項18の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリンタに出力するデータフロー可視化手段を備えたものである。

【0058】請求項19の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデ

ータフロー可視化手段と、プログラミングにおけるデータの演算順序の連續性を判定する連續性判定手段とを備えたものである。

【0059】請求項20の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段と、プログラミングにおけるデータの演算順序の連續性を判定する連續性判定手段とを備えたものである。

【0060】請求項21の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段と、出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示する演算順序／演算タイミング同時表示手段とを備えたものである。

【0061】請求項22の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段と、出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示する演算順序／演算タイミング同時表示手段とを備えたものである。

【0062】請求項23の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段と、元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上に変更箇所のみ追加出力する変更箇所追加出力手段とを備えたものである。

【0063】請求項24の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段と、元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上に変更箇所のみ追加出力する変更箇所追加出力手段とを備えたものである。

【0064】請求項25の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段と、元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図上での変更箇所がない場合に修正出力処を行なわない修正出

力制御手段とを備えたものである。

【0065】請求項26の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化するデータフロー可視化手段と、元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図上での変更箇所がない場合に修正出力処理を行なわない修正出力制御手段とを備えたものである。

【0066】請求項27の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリントに出力するデータフロー可視化手段と、プログラミングにおけるデータの演算順序の連続性を判定する連続性判定手段とを備えたものである。

【0067】請求項28の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリントに出力するデータフロー可視化手段と、プログラミングにおけるデータの演算順序の連続性を判定する連続性判定手段とを備えたものである。

【0068】請求項29の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリントに出力するデータフロー可視化手段と、出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示する演算順序／演算タイミング同時表示手段とを備えたものである。

【0069】請求項30の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリントに出力するデータフロー可視化手段と、出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示する演算順序／演算タイミング同時表示手段とを備えたものである。

【0070】請求項31の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリントに出力するデータフロー可視化手段と、元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上に変更箇所のみ追加出力する変更箇所追加出力手段とを備えたものである。

【0071】請求項32の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施する

プログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリントに出力するデータフロー可視化手段と、元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上に変更箇所のみ追加出力する変更箇所追加出力手段とを備えたものである。

【0072】請求項33の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリントに出力するデータフロー可視化手段と、元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図上での変更箇所がない場合に修正出力処理を行なわない修正出力制御手段とを備えたものである。

【0073】請求項34の発明に係るプログラミング装置は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリントに出力するデータフロー可視化手段と、元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図上での変更箇所がない場合に修正出力処理を行なわない修正出力制御手段とを備えたものである。

【0074】請求項35の発明に係るプログラミング装置は、プログラムを実行する装置と一体的に構成するようにしたものである。

【0075】

【作用】請求項1の発明におけるプログラミング装置は、CRTの画面上にポインティングデバイスにより作成されたデータフロー図をデータに対する共通の演算処理毎に分解し、その演算を実施するプログラム群を作成することで同一演算処理のデータに対し同一演算を一括して実行する一括高速処理を実現し、複数のデータに対する演算実行時のオーバーヘッド時間を削減する。

【0076】請求項2の発明における入出力データのファイル自動生成手段は、共通の演算処理を実施するプログラム群からそのプログラムに対し入出力データのファイルを別に生成することで、上記プログラムとスケジューリング手段入出力データのファイルとを分離し、上記プログラムを固定化することを可能にする。

【0077】請求項3の発明におけるスケジューリング手段は、共通の演算処理を実施するプログラムの実行タイミングをスケジューリングすることで人為的な処理順序の誤りを防止することを可能にする。

【0078】請求項4の発明におけるプログラミング装置は、共通の演算処理を実施するプログラム群からそのプログラムに対し入出力データのファイルを別に生成することで、上記プログラムと入出力データのファイルとを分離し、上記プログラムを固定化することを可能にする。

ると共に、共通の演算処理を実施するプログラムの実行タイミングをスケジューリングすることで人為的な処理順序の誤りを防止することを可能にする。

【0079】請求項5の発明における変更箇所修正出力手段は、元のデータフロー図上で発生した演算処理内容変更に対する処理を、出力されるプログラム上の変更箇所に対応する部分のみ修正し出力することで、元のデータフロー図上で発生した演算処理内容変更による時間や資料の無駄を省くことを可能にする。

【0080】請求項6の発明における変更箇所修正出力手段は、元のデータフロー図上で発生した演算処理内容変更に対する処理を、入出力データファイル上の変更箇所のみ修正し出力するので、元のデータフロー図上で発生した演算処理内容変更による上記入出力データファイルの作成し直しの時間や入出力データファイルの資料の無駄を省くことが可能となる。

【0081】請求項7の発明における変更箇所修正出力手段は、元のデータフロー図上で発生した演算処理内容変更に対する処理を、スケジューリングしたスケジュール上の変更箇所のみ修正し出力するので、元のデータフロー図上で発生した演算処理内容変更による上記スケジュールの作成し直しの時間やスケジュールに関する資料の無駄を省くことが可能となる。

【0082】請求項8の発明におけるプログラミング装置は、データに対する演算処理をブロック図化したデータフロー図を表わした手書きOCRシートを使用できるため、上記データフロー図が複数の者により作成可能となり、データフロー図の作成が容易となる。

【0083】請求項9の発明におけるデータフロー可視化手段は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化し、変更結果をCRT画面上で直接目視によりチェック可能にする。

【0084】請求項10の発明におけるプリンタ出力手段は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置しデータフロー図としてプリンタに出力するので、データフロー図が改造／修正され複雑になっても上記プリンタの用紙の長さには制限がないので、上記複雑になったデータフロー図の出力に支障を来さないように作用する。

【0085】請求項11の発明における連続性判定手段は、プログラミングにおけるデータの演算順序の連続性を判定することで、入力データと出力データの一貫性をチェックし、プログラムを変更したときの誤りを検出することを可能にする。

【0086】請求項12の発明における演算順序／演算タイミング同時表示手段は、出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示することでプログラムの修正の際の誤りのチェックを

容易にする。

【0087】請求項13の発明における変更箇所追加出力手段は、元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上で変更箇所のみ追加出しし、元のプログラム内で発生した演算対象データを変更したときに、それに従って行なわれる修正処理時間や修正される資料の無駄を省くことを可能にする。

【0088】請求項14の発明における修正出力制御手段は、元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図での変更箇所がない場合に修正出力処理を行なわないようにして、変更による修正処理に要する時間を省くことを可能にする。

【0089】請求項15の発明におけるプログラミング装置は、データフロー図とプログラム間の変換を双方向にし、さらにデータフロー可視化手段は、上記プログラムから変換したデータフロー図をCRTに出力することを可能にする。

【0090】請求項16の発明におけるプログラミング装置は、手書きOCRシートによるデータフロー図の入力を可能にすると共に、データフロー可視化手段は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置し、データフロー図としてCRT画面に可視化することを可能にする。

【0091】請求項17の発明におけるプログラミング装置は、データフロー図とプログラム間の変換を双方向にし、さらにデータフロー可視化手段は、上記プログラムから変換したデータフロー図をプリンタに出力することを可能にする。

【0092】請求項18の発明におけるプログラミング装置は、手書きOCRシートによるデータフロー図の入力を可能にすると共に、データフロー可視化手段は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置し、データフロー図としてプリンタに出力し可視化することを可能にする。

【0093】請求項19の発明におけるプログラミング装置は、データフロー図とプログラム間の変換を双方向にし、さらにデータフロー可視化手段は、上記プログラムから変換したデータフロー図をCRTに出力することを可能にし、連続性判定手段はプログラミングにおけるデータの演算順序の連続性を判定することで、入力データと出力データの一貫性をチェックし、プログラムを変更したときの誤りを検出することを可能にする。

【0094】請求項20の発明におけるプログラミング装置は、手書きOCRシートによるデータフロー図の入力を可能にすると共に、さらにデータフロー可視化手段は、上記プログラムから変換したデータフロー図をCRT

Tに出力することを可能にし、連續性判定手段はプログラミングにおけるデータの演算順序の連續性を判定することで、入力データと出力データの一貫性をチェックし、プログラムを変更したときの誤りを検出することを可能にする。

【0095】請求項21の発明におけるプログラミング装置は、データフロー図とプログラム間の変換を双方向にし、さらにデータフロー可視化手段は、上記プログラムから変換したデータフロー図をCRTに出力することを可能にし、演算順序／演算タイミング同時表示手段は、出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示することでプログラムの修正の際の誤りのチェックを容易にすることを可能にする。

【0096】請求項22の発明におけるプログラミング装置は、手書きOCRシートによるデータフロー図の入力を可能にすると共に、さらにデータフロー可視化手段は、上記プログラムから変換したデータフロー図をCRTに出力することを可能にし、演算順序／演算タイミング同時表示手段は、出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示することでプログラムの修正の際の誤りのチェックを容易にすることを可能にする。

【0097】請求項23の発明におけるプログラミング装置は、データフロー図とプログラム間の変換を双方向にし、さらにデータフロー可視化手段は、上記プログラムから変換したデータフロー図をCRTに出力することを可能にし、変更箇所追加出力手段は、元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上で変更箇所のみ追加出力し、元のプログラム内で発生した演算対象データを変更したときに、それに従って行なわれる修正処理時間や修正される資料の無駄を省くことを可能にする。

【0098】請求項24の発明におけるプログラミング装置は、手書きOCRシートによるデータフロー図の入力を可能にすると共に、さらにデータフロー可視化手段は、上記プログラムから変換したデータフロー図をCRTに出力することを可能にし、変更箇所追加出力手段は、元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上で変更箇所のみ追加出力し、元のプログラム内で発生した演算対象データを変更したときに、それに従って行なわれる修正処理時間や修正される資料の無駄を省くことを可能にする。

【0099】請求項25の発明におけるプログラミング装置は、データフロー図とプログラム間の変換を双方向にし、さらにデータフロー可視化手段は、上記プログラムから変換したデータフロー図をCRTに出力することを可能にし、修正出力制御手段は、元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、データフロー図上で

更箇所がない場合に修正出力処理を行なわないようにして、影響の小さな変更による修正処理に要する時間を省くことを可能にする。

【0100】請求項26の発明におけるプログラミング装置は、手書きOCRシートによるデータフロー図の入力を可能にすると共に、さらにデータフロー可視化手段は、上記プログラムから変換したデータフロー図をCRTに出力することを可能にし、修正出力制御手段は、元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図上での変更箇所がない場合に修正出力処理を行なわないようにして、影響の小さな変更による修正処理に要する時間を省くことを可能にする。

【0101】請求項27の発明におけるプログラミング装置は、データフロー図とプログラム間の変換を双方向にし、さらにデータフロー可視化手段は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置し、データフロー図としてプリンタに出力し可視化することを可能にし、連續性判定手段はプログラミングにおけるデータの演算順序の連續性を判定することで、入力データと出力データの一貫性をチェックし、プログラムを変更したときの誤りを検出することを可能にする。

【0102】請求項28の発明におけるプログラミング装置は、手書きOCRシートによるデータフロー図の入力を可能にすると共に、プリンタへのデータフロー図の出力を可能にし、さらに連続性判定手段はプログラミングにおけるデータの演算順序の連續性を判定することで、入力データと出力データの一貫性をチェックし、プログラムを変更したときの誤りを検出することを可能にする。

【0103】請求項29の発明におけるプログラミング装置は、データフロー図とプログラム間の変換を双方向にし、さらにデータフロー可視化手段は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置し、データフロー図としてプリンタに出力し可視化することを可能にし、演算順序／演算タイミング同時表示手段は、出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示することでプログラムの修正の際の誤りのチェックを容易にすることを可能にする。

【0104】請求項30の発明におけるプログラミング装置は、手書きOCRシートによるデータフロー図の入力を可能にすると共に、さらにデータフロー可視化手段は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置し、データフロー図としてプリンタに出力し可視化することを可能にし、演算順序／演算タイミング同時表示手段は、出力するデータフロー図上に演算順序情報あるいは演算

タイミング情報を同時に表示することでプログラムの修正の際の誤りのチェックを容易にすることを可能にする。

【0105】請求項31の発明におけるプログラミング装置は、データフロー図とプログラム間の変換を双方向にし、さらにデータフロー可視化手段は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置し、データフロー図としてプリンタに出力し可視化することを可能にし、変更箇所追加出力手段は、元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上で変更箇所のみ追加出力し、元のプログラム内で発生した演算対象データを変更したときに、それに従って行なわれる修正処理時間や修正される資料の無駄を省くことを可能にする。

【0106】請求項32の発明におけるプログラミング装置は、手書きO C Rシートによるデータフロー図の入力を可能にすると共に、さらにデータフロー可視化手段は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置し、データフロー図としてプリンタに出力し可視化することを可能にし、変更箇所追加出力手段は、元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上で変更箇所のみ追加出力し、元のプログラム内で発生した演算対象データを変更したときに、それに従って行なわれる修正処理時間や修正される資料の無駄を省くことを可能にする。

【0107】請求項33の発明におけるプログラミング装置は、データフロー図とプログラム間の変換を双方向にし、さらにデータフロー可視化手段は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置し、データフロー図としてプリンタに出力し可視化することを可能にし、修正出力制御手段は、元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図上での変更箇所がない場合に修正出力処理を行なわないようにすることで、影響の小さな変更による修正処理に要する時間を省くことを可能にする。

【0108】請求項34の発明におけるプログラミング装置は、手書きO C Rシートによるデータフロー図の入力を可能にすると共に、さらにデータフロー可視化手段は、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置し、データフロー図としてプリンタに出力し可視化することを可能にし、修正出力制御手段は、元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図上での変更箇所がない場合に修正出力処理を行なわないようにすることで、影響の小さな変更による修正処理に

要する時間を省くことを可能にする。

【0109】請求項35の発明におけるプログラミング装置は、プログラムを実行する装置と一体的に構成することで、作成／修正したプログラムをそのプログラムを実行する装置にロードするためのプログラムロード時間を削減する。

【0110】

【実施例】

実施例1. 以下、請求項1の発明の一実施例を図について説明する。図1は、請求項1の発明の一実施例によるプログラミング装置を示すブロック図であり、図27と同一または相当の部分については同一の符号を付し説明を省略する。図において34aはC R Tの画面上にポインティングデバイスにより作成されたデータフロー図をデータに対する共通の演算処理毎に分解し、その演算を実施するプログラム群を作成するプログラム作成手段である。

【0111】図2は発電所における「日誌」機能データフローを本実施例のプログラミング装置によりブロック図として表現したものである。図において51はブロック図として表現された「日誌」機能データフローの入力部、52は同様に演算部、53は出力部である。入力部51の「I W O O O 1」や「M O O 1」は演算処理の対象となるデータの変数名である。また54は平均処理、55は加算処理、56は積算処理、57は効率計算処理、58は端数処理、59は時間遅れ処理を示す。図3は出力されるプログラムの一例を示している。図において61は宣言文部、62は繰り返し部、63は演算処理部である。

【0112】次に動作について説明する。図4は、図2に示すブロック図として示した発電所における「日誌」機能データフローについて、操作入力を演算命令に変換する操作入力→演算命令変換処理を行なうプログラム作成手段34aの処理動作を示すフローチャートである。この処理動作は演算種類の決定を図2の画面の左から右、上から下へと走査することにより行なう。この結果、「平均処理」、「加算処理」、「積算処理」、「効率演算処理」、「端数処理」、「時間遅れ処理」の6種類の演算の存在を得ることになる(ステップS T 4 1)。

【0113】次に、演算処理毎の入出力変数とその数を得る(ステップS T 4 2)。図2の「日誌」機能データフローの「平均処理」に着目すると処理回数は「4」、入出力の組み合わせは(1) I W O O O 1とM O O 1、(2) I W O O O 4とM O O 3、(3) I W O O O 5とM O O 4、(4) I W O O O 6とM O O 5である。

【0114】次に図3に示す入出力変数の宣言文部61を作成する(ステップS T 4 3)。さらに次に、処理の繰り返し部62を作成する。このときステップS T 4 2で得た入出力数を基に作成する(ステップS T 4 4)。

そして次に、演算処理部63を作成する。演算処理部63は、図示していない演算メニューから該当する演算のソースコードを選択するという手段により作成される(ステップST45)。

【0115】このようにして「平均処理」についての演算のソース展開(プログラム作成)が行なわれる。なお、「加算処理」、「積算処理」、「効率計算処理」、「端数処理」、「時間遅れ処理」などの他の演算についても同様にしてプログラム展開を行なう。

【0116】このように本実施例によれば、同一演算対象データの一括高速処理が可能となる。

【0117】実施例2、以下、請求項2の発明の一実施例を図について説明する。図5は、本実施例のプログラミング装置の構成を示すブロック図であり、図1と同一または相当の部分については同一の符号を付し説明を省略する。図において34bは、共通の演算処理を実施するプログラム群を作成するのと同時に、そのプログラムのための入出力データファイルを自動生成する入出力データファイル自動生成手段である。

【0118】図6は、本実施例のプログラミング装置により作成されたプログラムを示し、図7は、入出力データをファイル化したときの入力データファイル、出力データファイル、処理個数を示している。図6において65は作成されたプログラム、66はファイルの読み込み処理を示している。図7において68は前記プログラムと同時に作成された入出力データセット数、69は入力データファイル、70は出力データファイルである。

【0119】このように、前記実施例1では入出力データを宣言文としてプログラム中に組み込んだのに対し、本実施例では入力データファイル、出力データファイル、処理個数をプログラムから分離して扱うことが出来、プログラム自身は固定したものにできる。

【0120】実施例3、以下、請求項3の発明の一実施例を図について説明する。図8は、本実施例のプログラミング装置の構成を示すブロック図であり、図1と同一または相当の部分については同一の符号を付し説明を省略する。図において34cは、共通の演算処理を実施するプログラム群を作成するのと同時に、スケジューリング手段によりプログラムの実行タイミングを自動的にスケジューリングするスケジューリング手段である。

【0121】図9は、本実施例のプログラミング装置により出力されるプログラムのスケジュール表である。図10は、本実施例の説明に使用するブロック図として示した発電所における「日誌」機能データフローである。図10において図2と同一または相当の部分については同一の符号を付し説明を省略する。図において72は「平均処理」である。

【0122】前記実施例1、実施例2のプログラミング装置では、プログラムを作成したりプログラムとそのプログラムと分離させた入出力データファイルを作成する

ように構成したが、プログラムの実行順序はオペレータが決める必要がある。このため、図2のブロック図として示した発電所における「日誌」機能データフローでは、「平均処理」→「加算処理」→「積算処理」→「端数処理」→「効率演算処理」→「時間遅れ処理」の順序で行なう必要がある。また、単純に同じ演算を全部一括して処理すると処理手順が逆転する可能性がある。たとえば、図10のブロック図として示した発電所における「日誌」機能データフローでは、「M008」を入力とし「M020」を出力とする平均処理72では、他の「平均処理」と同時に実行することが出来ない。

【0123】従って本実施例では、共通の演算処理を実施するプログラム群を作成するのと同時に、スケジューリング手段34cによりプログラムの実行タイミングを自動的にスケジューリングし、図9に示すスケジュール表を得る。そして、図示しないスケジュールプログラムがこのスケジュール表を参照し、各演算プログラムをそのスケジュールに従って起動する。

【0124】本実施例では、処理順序に関して発生することのある人為的な誤りが防止できる効果がある。

【0125】実施例4、なお、前記実施例1のプログラミング装置に入出力データファイル自動生成手段34bとスケジューリング手段34cを設けてもよく、同一演算対象データの一括高速処理が可能となると共に、入力データファイル、出力データファイル、処理個数をプログラムから分離することでプログラム自身を固定したものにでき、また処理順序に関して発生する人為的な誤りが防止できる効果がある。

【0126】実施例5、以下、請求項5の発明の一実施例を図について説明する。図11は、本実施例のプログラミング装置の構成を示すブロック図であり、図1と同一または相当の部分については同一の符号を付し説明を省略する。図において34dは、元のデータフロー図上で発生した演算処理内容変更に対する処理を、出力されるプログラム上の変更箇所に対応する部分のみ修正し出力する変更箇所修正出力手段である。

【0127】前記実施例1～実施例4では、プログラムの元データフロー図が一部修正された場合、プログラムやデータファイルやスケジュール表は最初から作成し直しとなる。変更が一部であるときには最初からの作り直しによる時間、資料の無駄を無くすために対象となる箇所のみの変更が望ましい。

【0128】図12は、ブロック図として示した発電所における「日誌」機能データフローを図2から図10に変更したときのスケジュール表の変化を示している。この場合スケジュール表の一部のみが変更されている。このように一部変更に対する変更箇所をたとえばスケジュール表上で最小にとどめることで、スケジュール表を最初から作り直す場合に比べ時間、資料などを削減できる。

【0129】実施例6. なお、前記実施例1のプログラミング装置に入出力データファイル自動生成手段34bと、元のデータフロー図上で発生した演算処理内容変更に対する処理を出力されるプログラム上の変更箇所に対応する入出力データファイルの部分のみ修正し出力する変更箇所修正出力手段を設けてもよく、同一演算対象データの一括高速処理が可能となると共に、一部変更に対する変更箇所をたとえば入出力データファイル上で最小にとどめることで、時間、資料などを削減できる。

【0130】実施例7. また、前記実施例1のプログラミング装置にスケジューリング手段34cとスケジューリングしたスケジュールまでの変更箇所のみ修正し出力する変更箇所修正出力手段を設けてもよく、処理順序に関して発生する人為的な誤りが防止できると共に、一部変更に対する変更箇所をスケジュール表上で最小にとどめることで、時間、資料などを削減できる。

【0131】実施例8. 以下、請求項8の発明の一実施例を図について説明する。図13は、本実施例のプログラミング装置の構成を示すブロック図であり、図1と同一または相当の部分については同一の符号を付し説明を省略する。図において81はOCR(Optical Character Reader)入力部であり、読み取った文字や図形を認識して、その認識結果を出力する機能を備えている。81aは手書きOCRシートである。

【0132】本実施例では、CRTとポインティングデバイスにより入力するデータフロー図をOCR入力部81により手書き手書きOCRシート81aにより入力できるので、CRT33を用いた場合にデータフロー図を複数のプログラム作成者により作成しようとするときはCRT33を増やす必要があるのに対し、このような必要がなくコストの増加を招来することがない。また、CRT33とポインティングデバイスを用いた場合に比べてデータフロー図の入力が簡単になり、データフロー図の作成が容易である。

【0133】実施例9. 実施例1から実施例8までは、データフロー図からの共通データ処理プログラムを作成するようにしたが、作成されたデータファイルやスケジュールプログラムが直接変更されたときにはその変更結果をチェックできない。変更されたプログラムですでに運用されている場合、元のデータとの整合性チェックは変更が人為的要素を含むため元のデータを理論的に変更後のプログラムに一致するように変更しプログラムを再度作成し直しても全く同一正を確保できないという理由で困難である。逆に共通データ処理プログラムからデータフロー図を作成する処理が必要である。従って、本実施例では共通データ処理プログラムからデータフロー図を作成する。

【0134】図14は本実施例のプログラミング装置の構成を示すブロック図であり、図1と同一または相当の

部分については同一の符号を付し説明を省略する。図において91は複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてCRT画面に可視化する演算→操作入力変換部(データフロー可視化手段)である。

【0135】図15は、共通処理プログラムから逆にデータフロー図を作成するデータフロー図作成処理手順を示すフローチャートである。以下、このフローチャートに基づいて共通処理プログラムから逆にデータフロー図を作成するデータフロー図作成処理手順を説明する。この場合、プログラム情報として実施例2および実施例3において説明した入出力ファイルおよびスケジュール表を用いる。

【0136】ステップST101は、スケジュール表から最初の演算要素を得た後、その演算要素の入力に対する出力を得て、さらにその出力を入力とする演算を求め繰り返す、ステップST101-1、ステップST101-2、ステップST101-3、ステップST101-4からなる一連の処理を行なうことで最初のデータフローを得るアルゴリズムを示している。ステップST102は、ステップST102-1、ステップST102-2からなる処理を行なうことで、ステップST101と同様にして同一の最初の演算要素に対する次のデータフローを得る手順を示している。このようにして最初の演算要素に対する全てのデータフローを得る(ステップST103)。その後さらにステップST104においてスケジュール表の2番目の演算要素に対し同様にデータフローを得る。この場合ステップST101からステップST103においてすでに処理した入力データ、演算処理の組み合わせは省く。このようにして全ての演算処理と全てのデータの連続性を処理した後、データフロー図として表示部33のCRTに出力する(ステップST105)。

【0137】このように本実施例では、作成されたデータファイルやスケジュールプログラムの変更に対し、その変更結果に応じたデータフロー図がCRT33に出力され表示されるため変更を直接チェックできる。

【0138】実施例10. 図16は本実施例のプログラミング装置の構成を示すブロック図であり、図14と同一または相当の部分については同一の符号を付し説明を省略する。図において92は複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリンタ93などの出力装置に可視化して出力する演算→操作入力変換部(データフロー可視化手段)である。

【0139】本実施例では、データフロー図が充分にCRT33に表現できないほどの広範囲な改造/修正が元のプログラムやスケジュール表に実施された場合、CRT33に出力するとその横方向の分解能に影響されて表

示が複雑化するので、CRT33ではないタイプライタやプロッタやプリンタなどの出力装置に用紙方向に展開されるデータフロー図を出力する。この結果、改造範囲が広範囲に及ぶような場合でも改造範囲に関係なく理解しやすいデータフロー図が得られる。

【0140】実施例11、前記実施例9と実施例10では、変更プログラムがデータに連続性があることを前提としているが、入力データと出力データの一貫性がないばあいには改造時の誤りとしてデータフロー図を出力する前に警報を出力させることができ。図17は、実施例9で示したアルゴリズムに入力データから最終出力データまでの一貫性についてのチェック機能を設けたときのフローチャートである。この一貫性についてのチェック機能を有するステップはステップST128により示される。このステップST128は、プログラミングにおけるデータの演算順序の連続性を判定する連続性判定手段である。

【0141】このように入出力データの一貫性をCRT33あるいはプリンタなどの出力装置にデータフロー図を出力する前にチェックすることにより無駄となる出力時間を省き、さらに改造したプログラムを実行する前の実行前チェックに使用できる。

【0142】実施例12、図18は本実施例のプログラミング装置の構成を示すブロック図であり、図16と同一または相当の部分については同一の符号を付し説明を省略する。図において92aは出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示する演算順序／演算タイミング同時表示手段である。

【0143】前記実施例9、実施例10、実施例11では、CRT33あるいはプリンタなどの出力装置により出力されたデータフロー図から入出力データの誤りのみしかチェックできない。図19は、本実施例のプログラミング装置により出力されるブロック図として示された発電所における「日誌」機能データフロー図であり、演算処理ブロックの右上にスケジュール表からの演算順序情報あるいは演算タイミング情報である実行順番が表示されている。

【0144】この結果、プログラムの修正時に演算順序指定の逆転が発生しているか否かを目視によりチェックできる。

【0145】実施例13、図20は本実施例のプログラミング装置の構成を示すブロック図であり、図16と同一または相当の部分については同一の符号を付し説明を省略する。図において92bは元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上に変更箇所のみ追加出力する変更箇所追加出力手段である。

【0146】実施例9および実施例10においては、元のプログラムに対する一部修正がなされた場合、出力さ

れるデータフロー図は最初から作成し直すことになる。変更が一部であるときには最初からの作成し直しによる時間、資料の無駄をなくすため、対象となる箇所のみの変更が望ましい。

【0147】図21は、図10に示す入出力情報の一ヶ所が変更されたときのスケジュール表を示している。この場合、スケジュール表のみが変更となっており、出力されるデータフロー図の入力データ「IW0005」が「IW0009」に変更される。このように一部変更に対する変更を変更箇所のみにして最小にとどめることで最初からの作成し直しによる時間、資料の無駄をなくすことができる。

【0148】実施例14、図22は、本実施例のプログラミング装置の構成を示すブロック図であり、図16と同一または相当の部分については同一の符号を付し説明を省略する。図において92cは元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図上で変更箇所がない場合に修正出力処理を行なわないようにする修正出力制御手段である。

【0149】実施例13は、元のプログラムの修正に対しデータフロー図の一部修正をおこなうように構成したが、修正がプログラムの入出力数や入出力変数の型に影響ないときにはデータフロー図の修正が不要であると判断し、修正を実行しない、とが望ましい。

【0150】図23は、データフロー図修正の必要性を判断する修正出力制御手段92cのアルゴリズムを示すフローチャートである。このフローチャートによれば、元のプログラムと比較した結果、変更箇所があると判断すると、修正実行開始後まず入出力データファイルに変更があるか否かをチェックし（ステップST151）、変更なければプログラムでの変更の有無をチェックする（ステップST152）。この結果プログラムでの変更があり、かつプログラム変更が入出力変数の宣言部でないことが判明すると処理を中止する（ステップST153）。

【0151】このように本実施例によれば、一部変更に対する変更が出力されるデータフロー図に影響ない場合には処理をスキップし、変更による時間の無駄を削減できる。

【0152】実施例15、なお、以上説明した実施例1～実施例8はデータフロー図からプログラムを作成する機能を有したプログラミング装置、実施例9～実施例14はプログラムから逆にデータフロー図を作成する機能を有したプログラミング装置として説明したが、プログラムを作成する機能とプログラムからデータフロー図を作成する機能を共に備えるようにしてもよい。

【0153】図24は、本実施例のプログラミング装置を示すブロック図であり、操作入力→演算命令変換部34は、実施例1のプログラム作成手段34aまたは実施

例2の出入力データファイル自動生成手段34bまたは実施例3のスケジューリング手段34cまたは実施例5の変更箇所修正出力手段34dまたはこれらプログラム作成手段34a、出入力データファイル自動生成手段34b、スケジューリング手段34c、変更箇所修正出力手段34dの組み合わせられた機能を有している。

【0154】演算→操作入力変換部91は、実施例9において説明したデータフロー可視化手段としての機能を有している。

【0155】このように本実施例では、プログラムとデータフロー図とを互に双方向で変換することが出来、プログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することが出来、プログラム作成の際の操作性が向上する。

【0156】実施例16、なお、実施例15におけるデータフロー図を入力するためのポインティングデバイスの代りに実施例8で説明したOCR入力部81を用いてもよい。

【0157】実施例17、また、実施例15における演算→操作入力変換部91が有する実施例9において説明したデータフロー可視化手段としての機能の代りに、実施例10に示す共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリンタ93などの出力装置に可視化して出力する演算→操作入力変換部92を用いてもよい。

【0158】実施例18、なお、実施例17におけるデータフロー図を入力するためのポインティングデバイスの代りに実施例8で説明したOCR入力部81を用いてもよい。

【0159】実施例19、また、実施例15のプログラミング装置に実施例11に示した連続性判定手段を付加するよう構成してもよい。

【0160】実施例20、なお、実施例19におけるデータフロー図を入力するためのポインティングデバイスの代りに実施例8で説明したOCR入力部81を用いてもよい。

【0161】実施例21、また、実施例15のプログラミング装置に実施例12に示した演算順序／演算タイミング同時表示手段92aを付加するよう構成してもよい。

【0162】実施例22、なお、実施例21におけるデータフロー図を入力するためのポインティングデバイスの代りに実施例8で説明したOCR入力部81を用いてもよい。

【0163】実施例23、また、実施例15のプログラミング装置に実施例13に示した変更箇所追加出力手段92bを付加するよう構成してもよい。

【0164】実施例24、なお、実施例23におけるデータフロー図を入力するためのポインティングデバイス

の代りに実施例8で説明したOCR入力部81を用いてもよい。

【0165】実施例25、また、実施例15のプログラミング装置に実施例14に示した修正出力制御手段92cを付加するよう構成してもよい。

【0166】実施例26、なお、実施例25におけるデータフロー図を入力するためのポインティングデバイスの代りに実施例8で説明したOCR入力部81を用いてもよい。

【0167】実施例27、また、実施例17のプログラミング装置に実施例11に示した連続性判定手段を付加するよう構成してもよい。

【0168】実施例28、なお、実施例27におけるデータフロー図を入力するためのポインティングデバイスの代りに実施例8で説明したOCR入力部81を用いてもよい。

【0169】実施例29、また、実施例17のプログラミング装置に実施例12に示した演算順序／演算タイミング同時表示手段92aを付加するよう構成してもよい。

【0170】実施例30、なお、実施例29におけるデータフロー図を入力するためのポインティングデバイスの代りに実施例8で説明したOCR入力部81を用いてもよい。

【0171】実施例31、また、実施例17のプログラミング装置に実施例13に示した変更箇所追加出力手段92bを付加するよう構成してもよい。

【0172】実施例32、なお、実施例31におけるデータフロー図を入力するためのポインティングデバイスの代りに実施例8で説明したOCR入力部81を用いてもよい。

【0173】実施例33、また、実施例17のプログラミング装置に実施例14に示した修正出力制御手段92cを付加するよう構成してもよい。

【0174】実施例34、なお、実施例33におけるデータフロー図を入力するためのポインティングデバイスの代りに実施例8で説明したOCR入力部81を用いてもよい。

【0175】実施例35、また、以上説明した実施例ではプログラミング装置はプログラムを実行する部分とは異なるものとして説明したが、プログラム実行装置と同一装置内でソフトウェアにより構成するようにしてもよい。図25は、本実施例のプログラミング装置の構成を示す概念図である。

【0176】本実施例によれば、プログラム実行装置と同一装置内でソフトウェアにより実現することができるハードウェアが削減できると共に、作成／修正後のプログラムロード時間を削減できる効果がある。

【0177】

【発明の効果】以上のように、請求項1の発明によれ

ば、C R Tの画面上に作成されたデータフロー図をデータに対する共通の演算処理毎に分解し、その演算を実施するプログラム群を作成するように構成したので、一括高速処理を実現し、複数のデータに対する演算実行時のオーバーヘッド時間を削減できる効果がある。

【0178】請求項2の発明によれば、共通の演算処理を実施するプログラム群を作成するのと同時に、そのプログラムのための入出力データファイルを自動生成するように構成したので、プログラムと入出力データファイルとを分離できプログラムを固定化できる効果がある。

【0179】請求項3の発明によれば、プログラムの実行タイミングを自動的にスケジューリングするように構成したので、人為的な処理順序の誤りを防止できる効果がある。

【0180】請求項4の発明によれば、プログラムのための入出力データファイルを自動生成する入出力データファイル自動生成手段および、上記プログラムの実行タイミングを自動的にスケジューリングするスケジューリング手段を備えるように構成したので、プログラムと入出力データファイルとを分離できプログラムを固定化できと共に人為的な処理順序の誤りを防止でき信頼性の高いプログラミング装置が得られる効果がある。

【0181】請求項5の発明によれば、元のデータフロー図上で発生した演算処理内容変更に対する処理を、出力されるプログラム上の変更箇所に対応する部分のみ修正し出力するように構成したので、元のデータフロー図上で発生した演算処理内容変更による時間や資料の無駄を省くことができる効果がある。

【0182】請求項6の発明によれば、元のデータフロー図上で発生した演算処理内容変更に対する処理を、入出力データファイル上の変更箇所のみ修正し出力するように構成したので、元のデータフロー図上で発生した演算処理内容変更による時間や資料の無駄を省くことができる効果がある。

【0183】請求項7の発明によれば、元のデータフロー図上で発生した演算処理内容変更に対する処理を、スケジューリング手段によりスケジューリングしたスケジュール上での変更箇所のみ修正し出力するように構成したので、元のデータフロー図上で発生した演算処理内容変更による時間や資料の無駄を省くことができる効果がある。

【0184】請求項8の発明によれば、データに対する演算処理をブロック図化したデータフロー図を表わした手書きO C Rシートを使用できるように構成したので、データフロー図が複数の者により作成可能となり、データフロー図の作成が容易となる効果がある。

【0185】請求項9の発明によれば、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてC R T画面に可視化するように構成したので、変

更結果をC R T画面上で直接目視によりチェックできる効果がある。

【0186】請求項10の発明によれば、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてプリンタに出力するように構成したので、C R Tに出力する場合のようにデータフロー図を出力する際の表示範囲の制限に関する障害を排除できる効果がある。

【0187】請求項11の発明によれば、プログラミングにおけるデータの演算順序の連続性を判定するように構成したので、入力データと出力データの一貫性をチェックでき、プログラムを変更したときの誤りを検出できる効果がある。

【0188】請求項12の発明によれば、出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示するように構成したので、プログラムの修正の際の誤りのチェックを容易にする効果がある。

【0189】請求項13の発明によれば、元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上に変更箇所のみ追加出力するように構成したので、元のプログラム内で発生した演算対象データを変更したときに、それに従って行なわれる修正処理時間や修正される資料の無駄を省くことができる効果がある。

【0190】請求項14の発明によれば、元のプログラム内において発生した変更が、出力されるデータフロー図上での変更にならない場合に修正出力処理を行なわないように構成したので、変更による修正処理に要する時間を省き、修正処理に要する時間を削減できる効果がある。

【0191】請求項15の発明によれば、データフロー図とプログラム間の変換を双方向にすると共に、プログラムから変換したデータフロー図をC R Tに出力できるように構成したので、プログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性が向上する効果がある。

【0192】請求項16の発明によれば、手書きO C Rシートを使用できるようにすると共に、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置することでデータフロー図としてC R T画面に可視化するように構成したので、データフロー図が複数の者により作成可能となり、データフロー図の作成が容易となると共に、プログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性が向上する効果がある。

【0193】請求項17の発明によれば、データフロー

図とプログラム間の変換を双方向にし、さらに上記プログラムから変換したデータフロー図をプリンタに出力できるように構成したので、プログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性が向上する効果がある。

【0194】請求項18の発明によれば、手書きO C Rシートを使用できるようにすると共に、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置し、データフロー図としてプリンタに出力し可視化するように構成したので、データフロー図が複数の者により作成可能となり、データフロー図の作成が容易となると共に、プログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性が向上する効果がある。

【0195】請求項19の発明によれば、データフロー図とプログラム間の変換を双方向にすると共に、上記プログラムから変換したデータフロー図をC R Tに出力でき、さらにプログラミングにおけるデータの演算順序の連続性を判定できるように構成したので、操作性が向上すると共に入力データと出力データの一貫性がチェックでき、プログラムを変更したときの誤りを検出できる効果がある。

【0196】請求項20の発明によれば、手書きO C Rシートを使用できるようにすると共に、プログラムから変換したデータフロー図をC R Tに出力でき、さらにプログラミングにおけるデータの演算順序の連続性を判定することで、入力データと出力データの一貫性をチェックできるように構成したので、データフロー図が複数の者により作成可能となり、データフロー図の作成が容易となると共にプログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性が向上し、またさらにプログラムを変更したときの誤りを検出できる効果がある。

【0197】請求項21の発明によれば、データフロー図とプログラム間の変換を双方向にすると共に、上記プログラムから変換したデータフロー図をC R Tに出力でき、さらに出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示するように構成したので、プログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性が向上し、またさらにプログラムの修正の際の誤りのチェックを容易にする効果がある。

【0198】請求項22の発明によれば、手書きO C Rシートを使用できるようにすると共に、プログラムから

変換したデータフロー図をC R Tに出力でき、さらに出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示するように構成したので、データフロー図が複数の者により作成可能となり、データフロー図の作成が容易となると共にプログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することが出来、プログラム作成の際の操作性が向上し、またさらにプログラムの修正の際の誤りのチェックを容易にする効果がある。

【0199】請求項23の発明によれば、データフロー図とプログラム間の変換を双方向にすると共に、上記プログラムから変換したデータフロー図をC R Tに出力でき、さらに元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上で変更箇所のみ追加出力するように構成したので、プログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性が向上し、またさらに元のプログラム内で発生した演算対象データを変更したときに、それに従って行なわれる修正処理時間や修正される資料の無駄を省くことができる効果がある。

【0200】請求項24の発明によれば、手書きO C Rシートを使用できるようにすると共に、プログラムから変換したデータフロー図をC R Tに出力でき、さらに元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上で変更箇所のみ追加出力するように構成したので、データフロー図が複数の者により作成可能となり、データフロー図の作成が容易となると共にプログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性が向上し、またさらに元のプログラム内で発生した演算対象データを変更したときに、それに従って行なわれる修正処理時間や修正される資料の無駄を省くことができる効果がある。

【0201】請求項25の発明によれば、データフロー図とプログラム間の変換を双方向にすると共に、上記プログラムから変換したデータフロー図をC R Tに出力でき、さらに元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれ、出力されるデータフロー図上での変更箇所がない場合に修正出力処理を行なわないように構成したので、プログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性が向上し、またさらに影響の小さな変更による修正処理に要する時間を省くことができる効果がある。

【0202】請求項26の発明によれば、手書きO C R

シートを使用できるようにすると共に、プログラムから変換したデータフロー図をC R Tに出力でき、さらに元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図上での変更箇所がない場合に修正出力処理を行なわないように構成したので、データフロー図が複数の者により作成可能となり、データフロー図の作成が容易となると共にプログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性が向上し、またさらに影響の小さな変更による修正処理に要する時間を省くことが出来る効果がある。

【0203】請求項27の発明によれば、データフロー図とプログラム間の変換を双方向にすると共に、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置し、データフロー図としてプリンタに出力し可視化し、さらにプログラミングにおけるデータの演算順序の連続性を判定するように構成したので、プログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性を向上させることができ、さらに入力データと出力データの一貫性をチェックできるのでプログラムを変更したときの誤りを検出できる効果がある。

【0204】請求項28の発明によれば、手書きO C Rシートを使用できるようにすると共に、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置し、データフロー図としてプリンタに出力し可視化し、さらにプログラミングにおけるデータの演算順序の連続性を判定するように構成したので、データフロー図が複数の者により作成可能となり、データフロー図の作成が容易となると共にプログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性を向上させることができ、またさらに入力データと出力データの一貫性をチェックできるのでプログラムを変更したときの誤りを検出できる効果がある。

【0205】請求項29の発明によれば、データフロー図とプログラム間の変換を双方向にすると共に、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置しデータフロー図としてプリンタに出力でき、さらに、出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示するように構成したので、プログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性を向上させ

することができ、さらにプログラムの修正の際の誤りのチェックを容易にできる効果がある。

【0206】請求項30の発明によれば、手書きO C Rシートを使用できるようにすると共に、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置しデータフロー図としてプリンタに出力でき、さらに出力するデータフロー図上に演算順序情報あるいは演算タイミング情報を同時に表示するように構成したので、データフロー図が複数の者により作成可能となり、データフロー図の作成が容易となると共にプログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性を向上させることができ、またさらにプログラムの修正の際の誤りのチェックが容易にできる効果がある。

【0207】請求項31の発明によれば、データフロー図とプログラム間の変換を双方向にすると共に、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置し、データフロー図としてプリンタに出力できるようにし、さらに元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上で変更箇所のみ追加出力するように構成したので、プログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性を向上させることができ、さらに元のプログラム内で発生した演算対象データを変更したときに、それに伴う修正処理時間や修正される資料の無駄を省くことができる効果がある。

【0208】請求項32の発明によれば、手書きO C Rシートを使用できるようにすると共に、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置し、データフロー図としてプリンタに出力でき、さらに元のプログラム内で発生した演算対象データの変更に対する処理を出力データフロー図上で変更箇所のみ追加出力するように構成したので、データフロー図が複数の者により作成可能となり、データフロー図の作成が容易となると共にプログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性を向上させることができ、さらに元のプログラム内で発生した演算対象データを変更したときに、それに従って行なわれる修正処理時間や修正される資料の無駄を省くことができる効果がある。

【0209】請求項33の発明によれば、データフロー図とプログラム間の変換を双方向にすると共に、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置し、データフロー

図としてプリンタに出力でき、さらに元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図上の変更箇所がない場合には修正出力処理を行なわないように構成したので、プログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性を向上させることができ、さらに影響の小さな変更による修正処理に要する時間を省くことができる効果がある。

【0210】請求項34の発明によれば、手書きOCRシートを使用できるようにすると共に、複数のデータに対する共通の演算処理を実施するプログラム群をデータの加工演算の順序に従い配置し、データフロー図としてプリンタに出力でき、さらに元のプログラム内において発生した変更が演算処理の内容変更のみについて行なわれていることを検出し、出力されるデータフロー図上の変更箇所がない場合に修正出力処理を行なわないように構成したので、データフロー図が複数の者により作成可能となり、データフロー図の作成が容易となると共にプログラムの作成／変更が容易となり、プログラム作成上の誤りの発生を削減でき、プログラムとデータフロー図の両方で確認することができ、プログラム作成の際の操作性を向上させることができ、またさらに影響の小さな変更による修正処理に要する時間を省くことができる効果がある。

【0211】請求項35の発明によれば、プログラムを実行する装置と一体的に構成したので、プログラムロード時間が削減できる効果がある。

#### 【図面の簡単な説明】

【図1】請求項1の発明の一実施例によるプログラミング装置を示すブロック図である。

【図2】請求項1の発明の一実施例によるプログラミング装置の表示部に表示された発電所における「日誌」機能データフロー図である。

【図3】請求項1の発明の一実施例によるプログラミング装置の表示部に表示された発電所における「日誌」機能データフロー図に対応したプログラムを示す説明図である。

【図4】請求項1の発明の一実施例によるプログラミング装置のプログラム作成手段の処理動作を示すフローチャートである。

【図5】請求項2の発明の一実施例によるプログラミング装置を示すブロック図である。

【図6】請求項2の発明の一実施例によるプログラミング装置により作成されたプログラムを示す説明図である。

【図7】請求項2の発明の一実施例によるプログラミング装置による入出力データをファイル化したときの入力データファイル、出力データファイル、処理個数を示す

説明図である。

【図8】請求項3の発明の一実施例によるプログラミング装置を示すブロック図である。

【図9】請求項3の発明の一実施例によるプログラミング装置におけるスケジュール表を示す説明図である。

【図10】請求項3の発明の一実施例によるプログラミング装置の表示部に表示された発電所における「日誌」機能データフロー図である。

【図11】請求項5の発明の一実施例によるプログラミング装置を示すブロック図である。

【図12】請求項5の発明の一実施例によるプログラミング装置におけるスケジュール表を示す説明図である。

【図13】請求項8の発明の一実施例によるプログラミング装置を示すブロック図である。

【図14】請求項9の発明の一実施例によるプログラミング装置を示すブロック図である。

【図15】請求項9の発明の一実施例によるプログラミング装置における共通処理プログラムからデータフロー図を作成するデータフロー図作成処理手順を示すフローチャートである。

【図16】請求項10の発明の一実施例によるプログラミング装置を示すブロック図である。

【図17】請求項11の発明の一実施例によるプログラミング装置における、図15に示すアルゴリズムに入力データから最終出力データまでの一貫性についてのチェック機能を設けたフローチャートである。

【図18】請求項12の発明の一実施例によるプログラミング装置を示すブロック図である。

【図19】請求項12の発明の一実施例によるプログラミング装置により出力されるブロック図として示された発電所における「日誌」機能データフロー図である。

【図20】請求項13の発明の一実施例によるプログラミング装置を示すブロック図である。

【図21】請求項13の発明の一実施例によるプログラミング装置の動作を説明するための発電所における「日誌」機能スケジュール表図や機能データフロー図である。

【図22】請求項14の発明の一実施例によるプログラミング装置を示すブロック図である。

【図23】請求項14の発明の一実施例によるプログラミング装置におけるデータフロー図修正の必要性を判断する修正出力制御手段のアルゴリズムを示すフローチャートである。

【図24】請求項15の発明の一実施例によるプログラミング装置を示すブロック図である。

【図25】請求項35の発明のプログラミング装置の構成を示す概念図である。

【図26】従来のプログラミング装置の論理回路図が表示された表示部の画面を示す説明図である。

【図27】従来のプログラミング装置の構成を示すプロ

ック図である。

【図28】従来のプログラミング装置のキー入力部のキー配列を示す説明図である。

【図29】従来のプログラミング装置における論理積(AND)や論理和(OR)、反転(NOT)の論理回路などとその論理回路などに対応した命令変換規則を示す論理記号説明図である。

【図30】従来のプログラミング装置における演算の処理順序を示す説明図である。

#### 【符号の説明】

31 操作入力部(ポインティングデバイス)

33 表示部(CRT)

34 a プログラム作成手段

34 b 入出力データファイル自動生成手段

34 c スケジューリング手段

34 d 変更箇所修正出力手段

81 OCR入力部

81 a 手書きOCRシート

91, 92 演算→操作入力変換部(データフロー可視化手段)

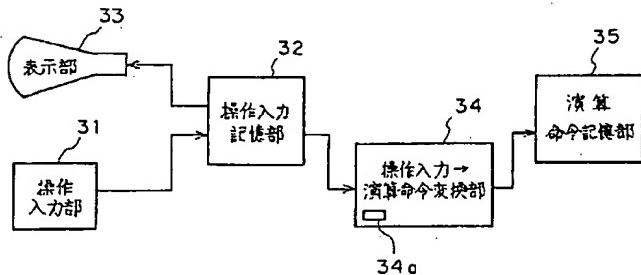
92 a 演算順序/演算タイミング同時表示手段

92 b 変更箇所追加出力手段

92 c 修正出力制御手段

93 プリンタ

【図1】



31: 操作入力部(ポインティングデバイス)

33: 表示部(CRT)

34 a: プログラム作成手段

【図3】

```

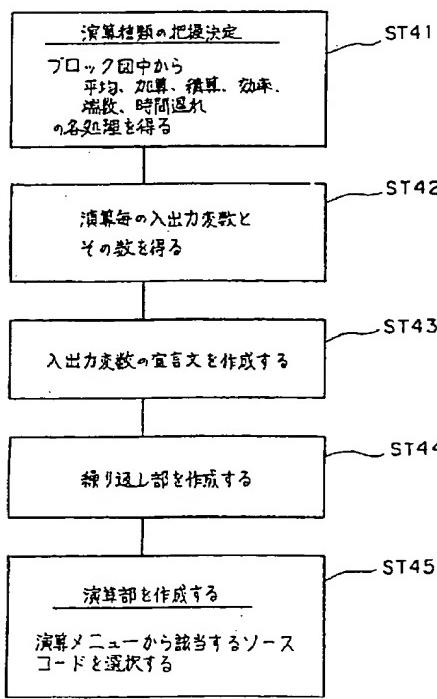
REAL SMOOTH
REAL IW(4)
REAL M(4)
REAL IW001, IW004, IW005, IW006
REAL M001, M003, M004, M005
EQUIVALENCE (IW(1), IW001)
EQUIVALENCE (IW(2), IW004)
EQUIVALENCE (IW(3), IW005)
EQUIVALENCE (IW(4), IW006)
EQUIVALENCE (M(1), M001)
EQUIVALENCE (M(2), M003)
EQUIVALENCE (M(3), M004)
EQUIVALENCE (M(4), M005)

DO 999 I=1, 4
M(I)=SMOOTH(IW(I)) ————— 63
999 END
  
```

61

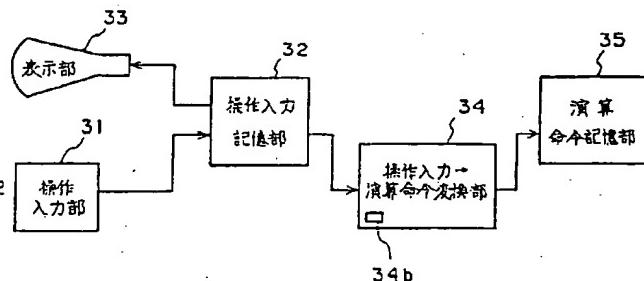
62

【図4】



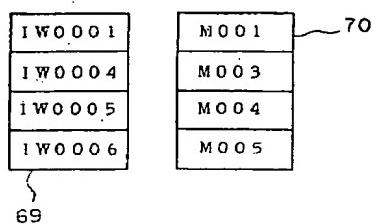
34 b: 入出力データファイル自動生成手段

【図5】

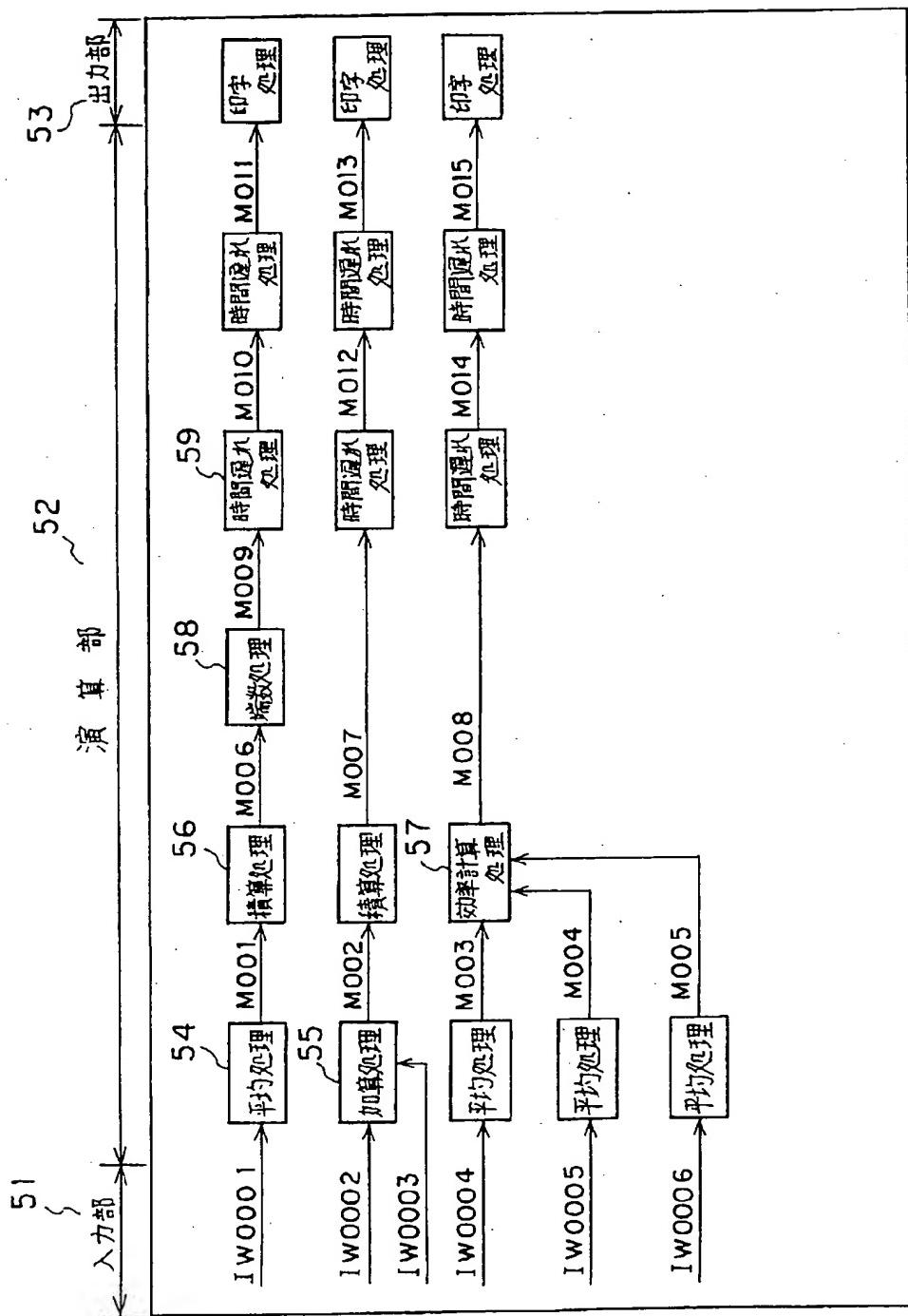


34 b: 入出力データファイル自動生成手段

【図7】



【図2】



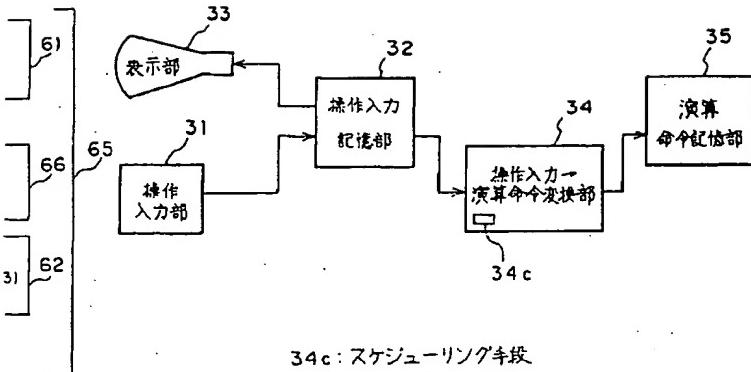
【図6】

```

REAL SMOOTH
REAL IW(900)
REAL M(900)

CALL GETF(1, 1, N)
CALL GETF(1, 2, IW)
DO 999 I=1, N
M(1) = SMOOTH(IW(1))—(23)
999 END
CALL OUTIF(1, 3, M)

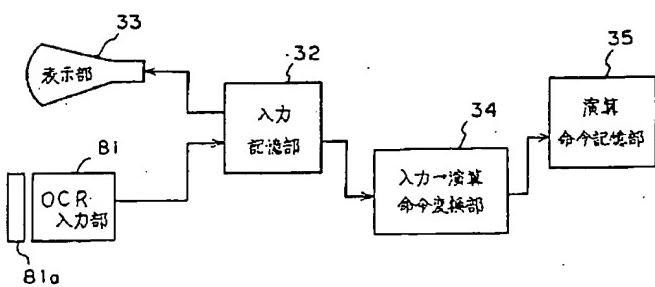
```



【図9】

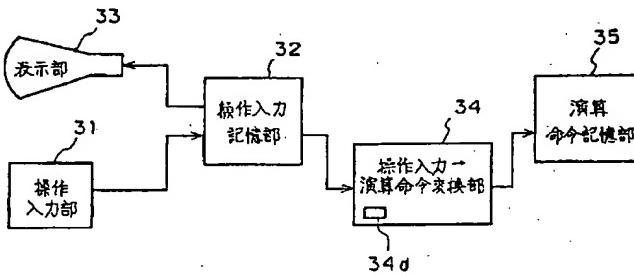
平均	4 セット	入力 1 = IW0001 入力 2 = IW0004 入力 3 = IW0005 入力 4 = IW0006	出力 1 = M001 出力 2 = M003 出力 3 = M004 出力 4 = M005
加算	1 セット	入力 1 = IW0002 入力 2 = IW0003	出力 1 = M002
積算	2 セット	入力 1 = M001 入力 2 = M002	出力 1 = M006 出力 2 = M007
効率	1 セット	入力 1 = M003 入力 2 = M004 入力 3 = M005	出力 1 = M008
端数	1 セット	入力 1 = M006	出力 1 = M009
平均	1 セット	入力 1 = M008	出力 1 = M020
時間遅れ	6 セット	入力 1 = M009 入力 2 = M007 入力 3 = M020 入力 4 = M010 入力 5 = M012 入力 6 = M014	出力 1 = M010 出力 2 = M012 出力 3 = M014 出力 4 = M011 出力 5 = M013 出力 6 = M015
印字	3 セット	入力 1 = M011 入力 2 = M013 入力 3 = M015	

【図13】



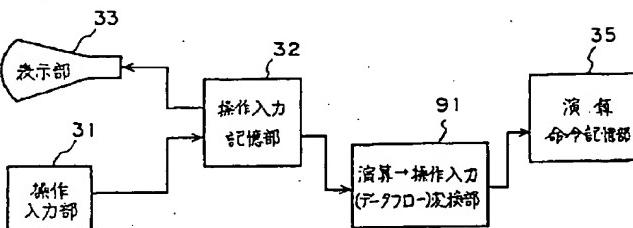
81a: 手書き OCRシート

【図11】



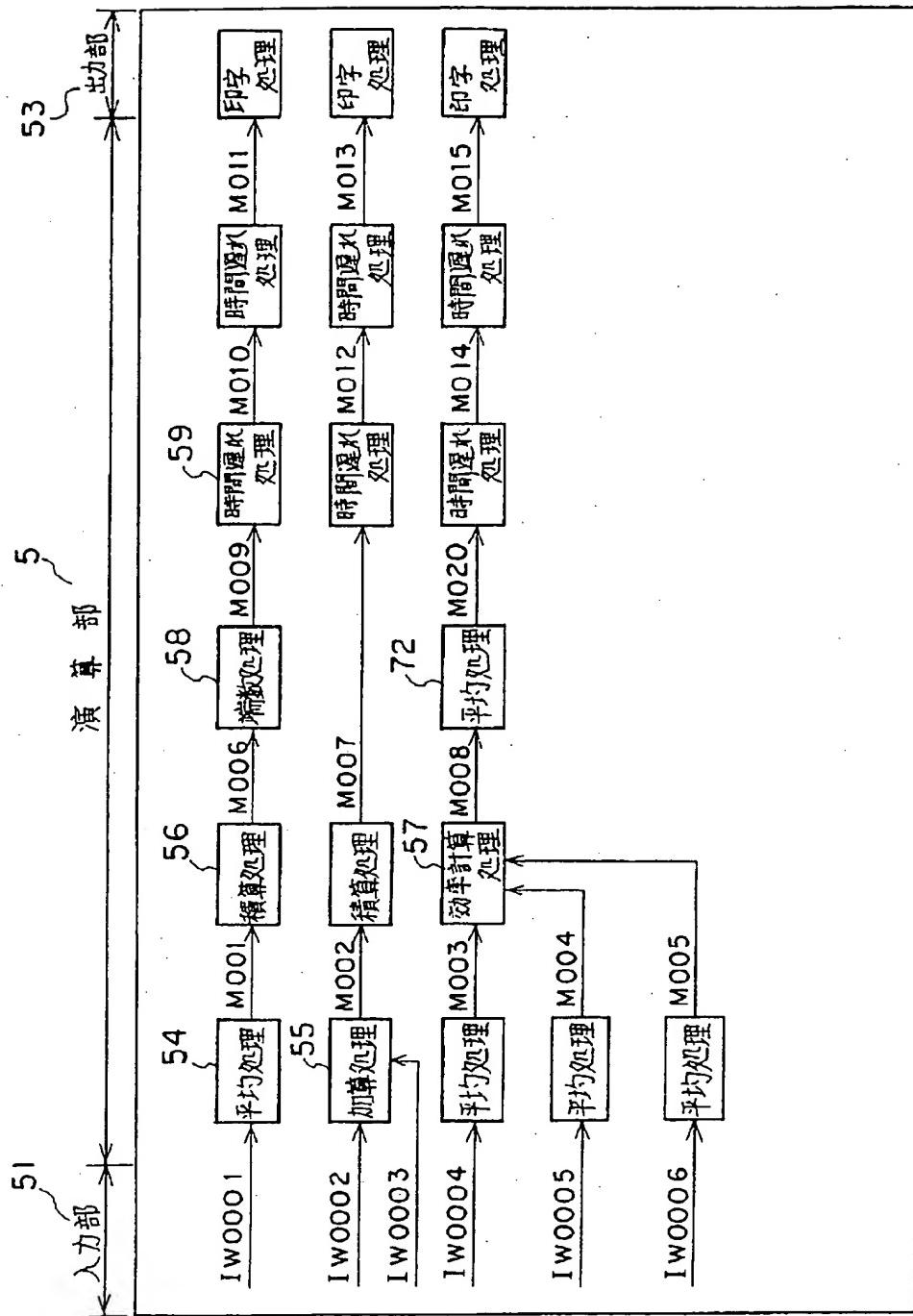
34d: 変更箇所修正出力手段

【図14】



91: 演算→操作入力交換部 (データフロー可視化手段)

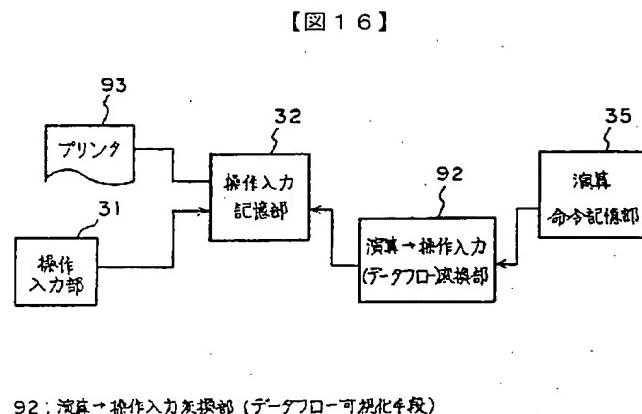
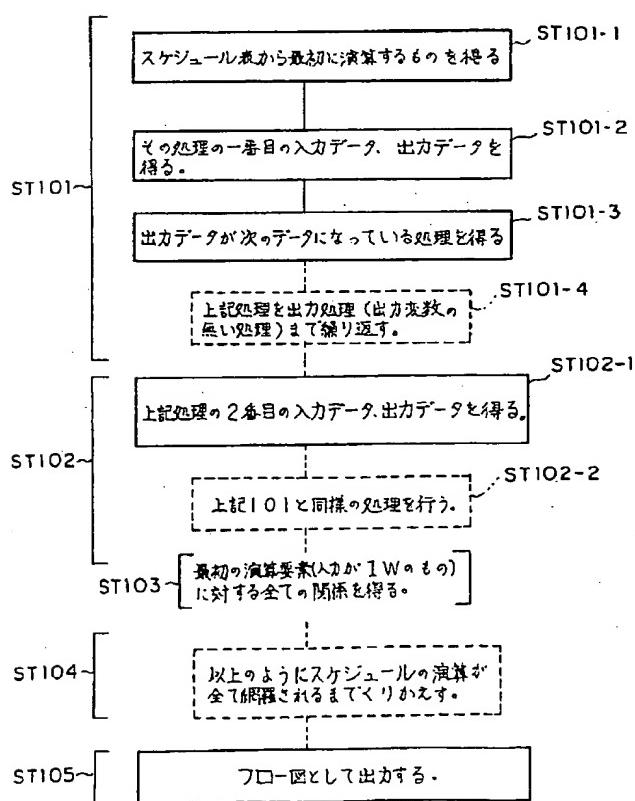
【図10】



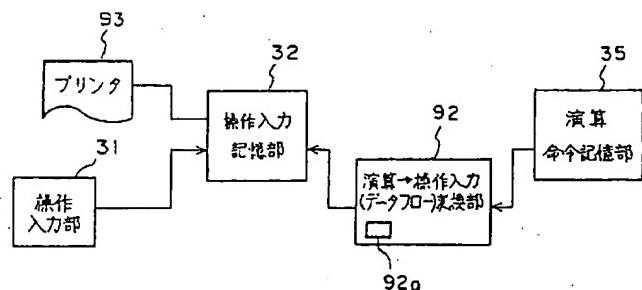
【図12】

平均	4セット	入力1 = M0001 入力2 = M0004 入力3 = M0005 入力4 = M0006	出力1 = M001 出力2 = M003 出力3 = M004 出力4 = M005		平均	4セット	入力1 = M0001 入力2 = M0004 入力3 = M0005 入力4 = M0006	出力1 = M001 出力2 = M003 出力3 = M004 出力4 = M005
加算	1セット	入力1 = M0002 入力2 = M0003	出力1 = M002	→	加算	1セット	入力1 = M0002 入力2 = M0003	出力1 = M002
積算	2セット	入力1 = M001 入力2 = M002	出力1 = M006 出力2 = M007		積算	2セット	入力1 = M001 入力2 = M002	出力1 = M006 出力2 = M007
効率	1セット	入力1 = M003 入力2 = M004 入力3 = M005	出力1 = M008		効率	1セット	入力1 = M003 入力2 = M004 入力3 = M005	出力1 = M008
掛け	1セット	入力1 = M006	出力1 = M009		掛け	1セット	入力1 = M008	出力1 = M009
時間差れ	6セット	入力1 = M009 入力2 = M007 入力3 = M020 入力4 = M010 入力5 = M012 入力6 = M014	出力1 = M010 出力2 = M012 出力3 = M014 出力4 = M011 出力5 = M013 出力6 = M015	↓	時間差れ	6セット	入力1 = M009 入力2 = M007 入力3 = M020 入力4 = M010 入力5 = M012 入力6 = M014	出力1 = M010 出力2 = M012 出力3 = M014 出力4 = M011 出力5 = M013 出力6 = M015
印字	3セット	入力1 = M011 入力2 = M013 入力3 = M015		↓	印字	3セット	入力1 = M011 入力2 = M013 入力3 = M015	

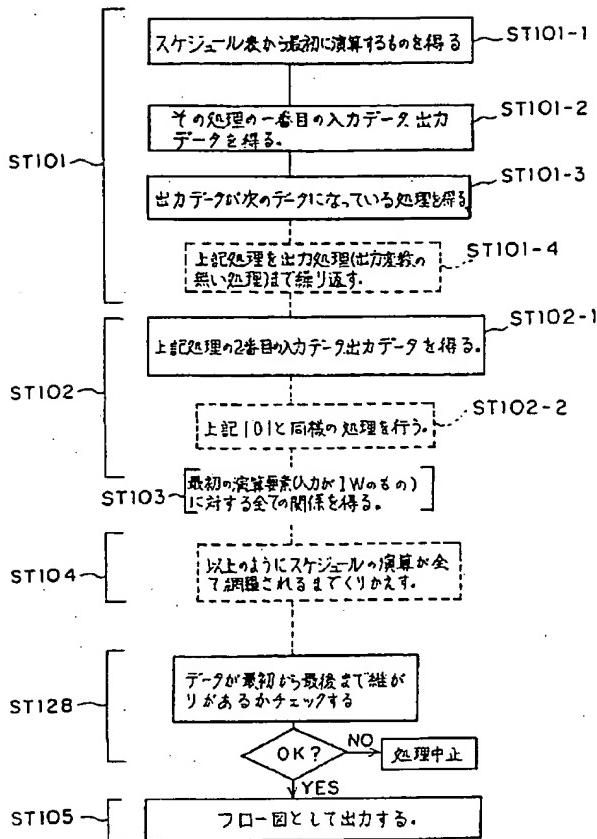
【図15】



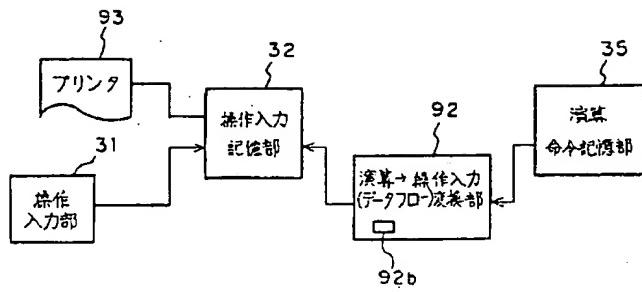
【図18】



【図17】

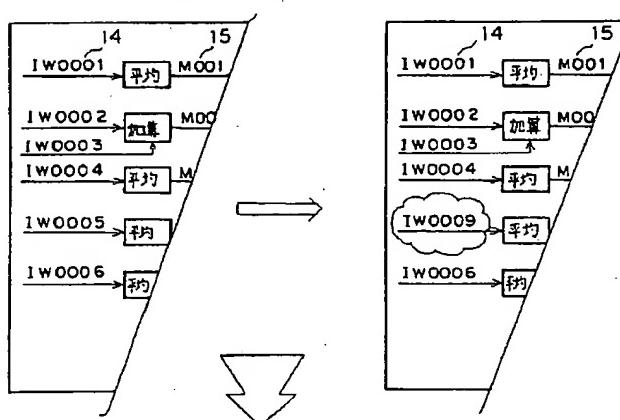


【図20】

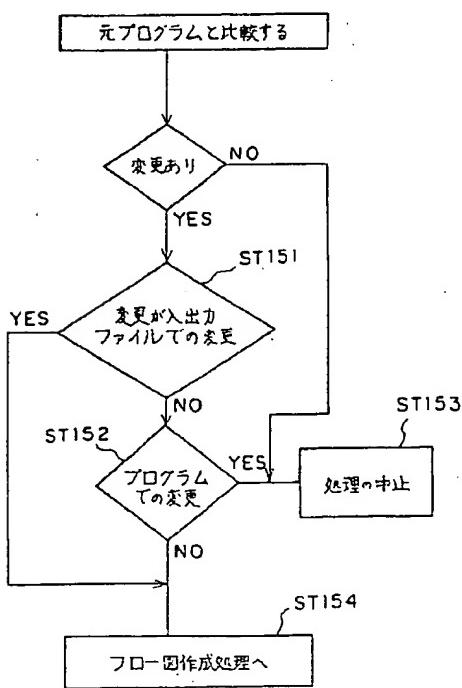


92b: 変更箇所追加出力手段

【図21】



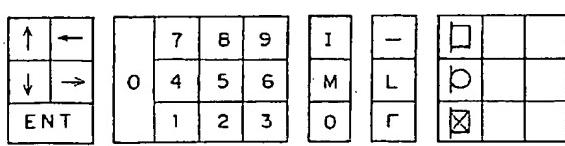
【図23】



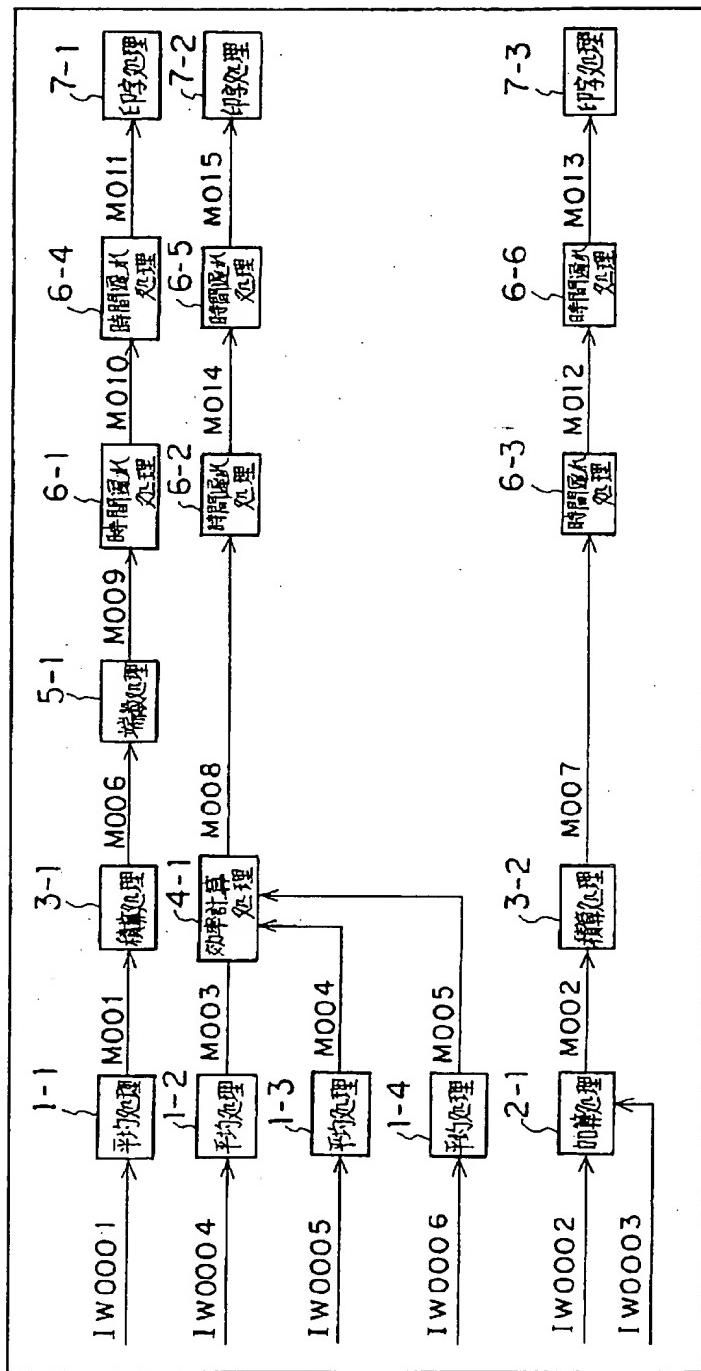
平均	4 セット	入力 1 = IW0001 入力 2 = IW0004 入力 3 = IW0005 入力 4 = IW0008 カ 1 = IW0002	出力 1 = M001 出力 2 = M003 出力 3 = M004 出力 4 = M005 カ 1 = M002 カ 2 = M003 カ 3 = M004 カ 4 = M005 カ 5 = M006
----	-------	--	--

平均	4 セット	入力 1 = IW0001 入力 2 = IW0004 <b>入力 3 = IW0009</b> 入力 4 = IW0008 カ 1 = IW0002	出力 1 = M001 出力 2 = M003 出力 3 = M004 出力 4 = M005 カ 1 = M002 カ 2 = M003 カ 3 = M004 カ 4 = M005 カ 5 = M006
----	-------	---	--

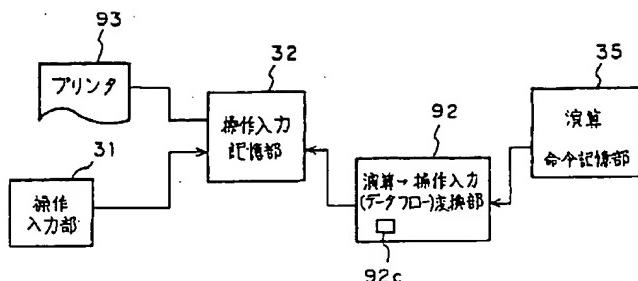
【図28】



【図19】

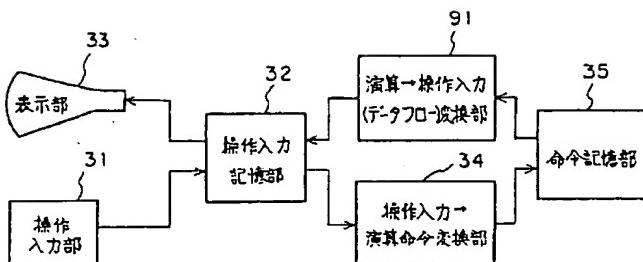


【図22】

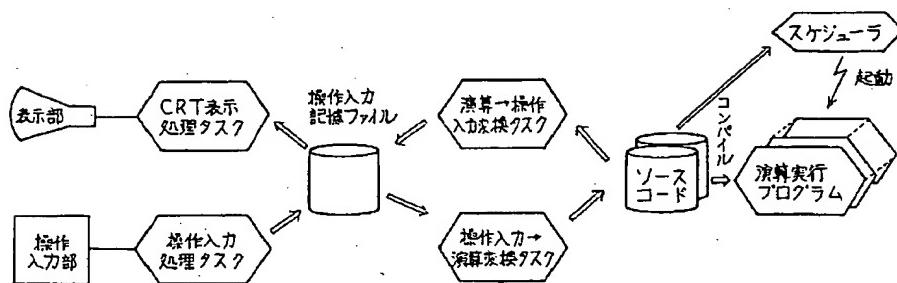


92c:修正出力制御手段

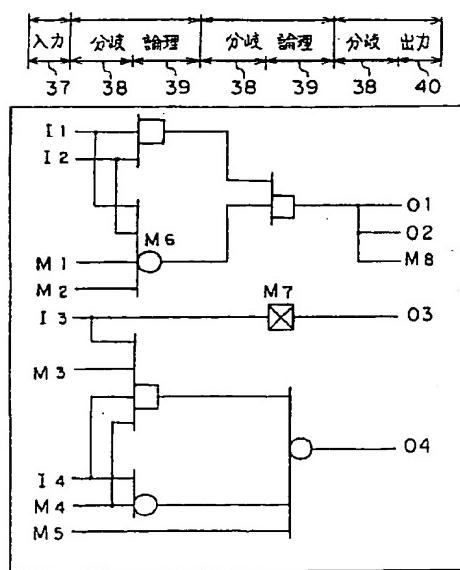
【図24】



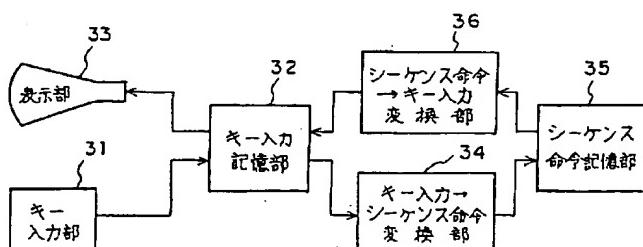
【図25】



【図26】



【図27】



【図29】

論理回路図		機能
ロジ 番号	論理	
m	i -	入力
$m+p$ $m+q$ $m+r$	Mi	論理積
$m+p$ $m+q$ $m+r$	Mi	論理和
m	Mi	否定
$m+p$ $m+r$	—	信号分岐
$m+p$ $m+r$	E	信号分岐
$m-p$ m	F	信号分岐
$m-p$ m	H	信号分岐
$m-p$ $m+r$	I	信号分岐
m	-Oi	出力

【図30】

